Cold Start to Improve Market Thickness on Online Advertising Platforms: Data-Driven Algorithms and Field Experiments

Zikun Ye[1],Dennis J. Zhang[2], Heng Zhang[3], Renyu Zhang[4], Xin Chen[1], Zhiwei Xu
[1] University of Illinois at Urbana-Champaign, Urbana, IL
[2] Washington University in St. Louis, St. Louis, MO
[3] Arizona State University, Tempe, AZ
[4] New York University Shanghai, Shanghai, China
zikunye2@illinois.edu, denniszhang@wustl.edu, hengzhang24@asu.edu, renyu.zhang@nyu.edu, xinchen@illinois.edu, rickyzhiwei@gmail.com

Cold start describes a commonly recognized challenge for online advertising platforms: with limited data, the machine learning system cannot accurately estimate the click-through rates (CTR) of new ads and, in turn, cannot efficiently price these new ads or match them with platform users. Traditional cold start algorithms often focus on improving the learning rates of CTR for new ads to improve short-term revenue, but unsuccessful cold start can prompt advertisers to leave the platform, decreasing the thickness of the ad marketplace. To address these issues, we build a data-driven optimization model that captures the essential trade-off between short-term revenue and long-term market thickness on the platform. Based on duality theory and bandit algorithms, we develop the Shadow Bidding with Learning (SBL) algorithms with a provable regret upper bound of $O\big(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}}\big)$, where K is the number of ads and d is the effective dimension of the underlying machine learning oracle for predicting CTR. Our proposed algorithms can be implemented in a real online advertising system with minimal adjustments. To demonstrate this practicality, we have collaborated with a large-scale video-sharing platform, conducting a novel, two-sided randomized field experiment to examine the effectiveness of our SBL algorithm. Our results show that the algorithm increased the cold start success rate by 61.62% while compromising short-term revenue by only 0.717%. Our algorithm has also boosted the platform's overall market thickness by 3.13% and its long-term advertising revenue by (at least) 5.35%. Our study bridges the gap between the bandit algorithm theory and the cold start practice, highlighting the value of well-designed cold start algorithms for online advertising platforms.

Key words: Cold Start Problem, Online Advertising, Contextual Bandit, Two-Sided Field Experiment

1. Introduction

With the rapid growth of internet technology and smartphone penetration, online advertising has become an enormous industry, with a substantial impact on the entire economy. The Interactive Advertising Bureau reports that online advertising revenue in the United States increased to $124.6 billion in 2019 (16% year-over-year growth rate compared to 2018, 19% average annual growth rate since 2010), 70% of which comes from mobile advertising.[1] Facebook, TikTok, and other large online

[1] See https://www.iab.com/insights/internet-advertising-revenue-fy2019-q12020/ for more details.

platforms monetize their gigantic user traffic primarily through online advertising. For example, in 2019, Facebook earned $69.7 billion revenue from advertising—98.53% of its total revenue.[2]

Online advertising systems, like many other Internet-based digitalized platforms, face a critical challenge called the cold start problem (see, e.g., Dave and Varma 2014, Choi et al. 2020). People have noted—both in the literature and in practice—that limited data history prevents online advertising platforms from accurately predicting the click-through rate (CTR) and the conversion rate (CVR) of new ads. While most of the existing literature focuses on improving the statistical properties of cold start algorithms—improving the learning rates for CTR and CVR—to maximize the short-run advertising revenue, we propose considering another important economic factor—market thickness—when designing cold start algorithms. Specifically, we observe that in practice, throughout a whole ad campaign, advertisers especially value an ad's performance in the first few days—a bad performance with few conversions (i.e., app installs, purchases) may lead the advertiser to remove the ad from the platform. Therefore, it is crucial that platforms help new ads perform well and economically win advertisers' loyalty, maintaining the thickness of the ad pool in the auction (see Section 3 for details) while learning the CTR and CVR of these new ads during cold start.

To illustrate that the performance of new ads during cold start could fundamentally impact the long-term behavior of these ads, we collaborate with a large-scale online video-sharing platform (referred to as Platform O hereafter), and plot in Figure 1 the relationship between the number of conversions per day during new ads' first three days on the platform (the cold start period; on the x-axis) and the retention rate of these new ads in the subsequent two weeks on Platform O.[3] Two key observations emerge from Figure 1: (1) the long-term retention rate of a new ad is positively correlated with its performance during the cold start, and (2) such positive correlation is flattened when the number of conversions reaches a threshold around 10. In other words, for an ad platform to have enough market thickness and, in turn, high long-run revenue, quickly accumulating the first few conversions of each new ad is essential.[4] Not only the ad retention dependent on the cold start, advertisers are also sensitive to whether their ads can obtain enough conversions during the cold start. On Platform O, advertisers will carefully monitor ad performance during the cold start; they may tighten the budget, reduce ad materials, or leave the platform if they are unsatisfied with the performance. Therefore, cold start performance will significantly impact both the thickness of the ad marketplace and the number of advertisers for an online advertising platform.

[2] See the financial report of Facebook: https://www.sec.gov/ix?doc=/Archives/edgar/data/1326801/000132680120000013/fb-12312019x10k.htm.

[3] The retention rate in these two weeks is defined as the number of ads that have exposure to users every day in these two weeks divided by the total number of ads. To protect the platform's identity and sensitive data, we re-scale the y-axis value to $[0, 1]$. The curve pattern remains the same if we vary the duration from one day to fourteen days.

[4] Obviously, Figure 1 shows only the correlation between new ads' conversions during the cold start and their long-run retention rates. In Appendix C.1, we also conduct extensive empirical analysis to provide causal evidence that gaining 10 conversions during the cold start will significantly boost ad retention by 15.03%.

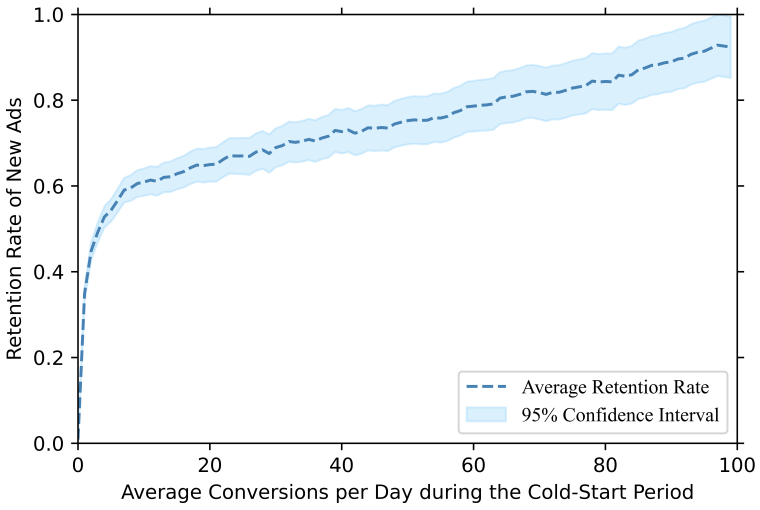


Figure 1 Retention Rate

On the other hand, to boost retention of new ads, platforms cannot simply provide more traffic to these new ads during cold start. This is because, as discussed above, the platform has less information about these new ads during cold start and, in turn, are less likely to efficiently match potential customers with these new ads. The inability to accurately predict CTR and CVR for new ads naturally brings up the exploration-exploitation trade-off between the short-term revenue generated by matching more mature ads (exploitation) and the long-term value from market thickness by matching more new ads (exploration). The fundamental trade-off in solving the cold start problem is to dynamically balance the long-term gain of successfully cold starting new ads and the short-term disutility from inefficiently matching new ads during cold start to optimize long-run ad revenue.

The main goal of this paper is to develop a new, theoretically sound and practically feasible end-to-end approach to solve the cold start problem when market thickness is important. To this end, we build a novel data-driven optimization model that integrates both the short-term revenue and the long-term cold start reward (defined as the long-term value from conversions during cold start to boost future market thickness) of an advertising platform. We develop a primal-dual-based multi-armed bandit (MAB) algorithm, denoted as the Shadow Bidding with Learning (SBL) algorithm, which adaptively adds a shadow bid to each new ad's bidding price. Our proposed algorithm adeptly bridges theory and practice: it has a provable performance guarantee *and* it could be straightforwardly implemented in the real-time bidding system of an online advertising platform with minimal adjustments. To demonstrate the practical value of our algorithm, we collaborated with a large-scale online video-sharing platform (Platform O) to conduct a large-scale randomized field experiment to evaluate our algorithm.[5] Our results show that the proposed algorithm sig-

[5] Platform O's area under the curve (AUC) of new-ad CTR prediction is 5.77% smaller than that of mature ads, a sizable gap for a large-scale platform that indicates it is more difficult to predict the CTR of new ads than that of mature

nificantly increases both the cold start reward of new ads and the long-term total revenue of the platform.

We summarize the main contributions of this paper as follows:

Optimization Model to Capture Cold Start and Market Thickness. Previous research on cold start in advertising has focused mainly on improving CTR and CVR prediction accuracy and/or learning rates (Choi et al. 2020). We are the first in the literature to consider the economic aspect of cold start, and to explicitly model cold start as an important lever for improving market thickness. We formulate the cold start problem for online advertising platforms as a data-driven optimization model, synthesizing the linear program and MAB models in an innovative fashion. We believe our new modeling assumptions are critical, because identifying promising ads with high retention and keeping the ads market-thick becomes notoriously challenging and important for not only the platform we work with but also other advertising platforms. Therefore, our modeling framework has the potential to empower other studies of the cold start problem for online advertising and recommender systems from an optimization perspective.

End-to-End Solution to the Cold Start Problem for Online Advertising. To the best of our knowledge, we are the first in the literature to provide an end-to-end implementation of a new algorithm to address the cold start problem for online advertising platforms when market thickness is important. We develop a novel SBL algorithm by embedding a linear program primal-dual framework into an ϵ-greedy contextual bandit algorithm. Though theoretically compelling, existing algorithms for general contextual bandits with concave objectives (e.g., Agarwal et al. 2014, Agrawal et al. 2016) are practically infeasible on real-world online advertising platforms. This is because these algorithms rely on an underlying *argmax oracle* (AMO), which is unavailable or computationally intractable in practice. Our proposed bandit algorithm bridges the gap between the learning theory and advertising cold start practice with a provable performance guarantee and straightforward implementation on real online advertising platforms. The algorithm leverages the dual variables of the cold start reward constraints for exploitation, the power of the advertising platform's underlying machine learning system to predict CTR and CVR, and an $\epsilon-$greedy exploration scheme, thus yielding a provable regret bound of $O\big(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}}\big)$, where K is the number of new ads and d is the effective dimension of the underlying machine learning oracle for predicting CTR and CVR. We also incorporate the dual mirror descent method into the SBL algorithm, which reduces its computational complexity without compromising the regret bound. Another compelling advantage of our SBL algorithm is that it enables us to minimally adjust the

ones. This prediction inaccuracy is amplified by the sparsity of conversions. Along these lines, Facebook recommends that its advertisers earmark enough budget for at least 50 conversions to successfully bring their ads out of the initial learning phase (i.e., the cold start period). See https://www.facebook.com/business/help/112167992830700?id=561906377587030.

real-time bidding system of an online advertising platform by simply adding a shadow bid for each ad (i.e., the dual variable of the cold start reward constraint) to its real-time bidding price.

Experimental Evaluation of Our Algorithm. We are the first in the literature to conduct two-sided randomized field experiments for bandit algorithms. In a general advertising cold start setting, the traditional one-sided experiment is invalidated by the violation of the Stable Unit Treatment Value Assumption (SUTVA) (see Section 5 for more discussion on this point) and, therefore, gives rise to estimation biases as high as 120%. Such violation of SUTVA is common in the experimental evaluation of algorithms and policies on e-commerce (e.g., Facebook Marketplace; see Ha-Thuc et al. 2020) and vacation-rentals (e.g., Airbnb; see Johari et al. 2020) platforms and has caused substantially biased estimations for experiments thereof. To address such a challenge, we design and implement a novel two-sided field experiment on Platform O. Under mild assumptions, the experiment restores SUTVA and enables us to causally estimate the value of our proposed algorithm in an unbiased fashion. The new experiment framework could be applied to evaluate other algorithms and policies of recommender systems on two-sided platforms. Based on our two-sided field experiments, we find that the proposed algorithm successfully increases the cold start success rate by 61.62%. Our experiment also demonstrates that the SBL algorithm increases the average retention time of the ads and, thus, market thickness by 3.13%. Moreover, we conduct comprehensive simulation studies which show that the total (long-term) advertising revenue of the entire platform also increases by at least 5.35% if the new cold start algorithm is applied, which translates to hundreds of millions of US dollars revenue boost per year for Platform O. In short, the two-sided experiments enable us to demonstrate that the SBL algorithm substantially improves the long-term revenue of an advertising platform.

In short, our study bridges the gap between the bandit algorithm theory and the cold start practice, highlighting the significant value of well-designed cold start algorithms for online advertising platforms. The rest of this paper is organized as follows. In Section 2, we position our paper in the relevant literature. Section 3 discusses the business practices on which we base our model. In Section 4, we propose our algorithms and analyze the regret bound. In Section 5, we introduce our field experiment setting. Section 6 reports our experimental results. Section 7 concludes. All proofs are relegated to the Online Appendices.

2. Literature Review

Our paper is primarily related to three streams of literature: cold start for online advertising, bandit algorithms, and field experiments on large-scale online platforms.

Estimating the CTR of new ads is a challenging problem, because there is very little data and information to provide reliable prediction (see, e.g., Dave and Varma 2014, Choi et al. 2020).

The sophisticated deep learning models developed in recent years are designed to better estimate the CTR of cold start items. For example, Zhou et al. (2018) propose the deep interest network, which incorporates data on users' historical behavior and interests to learn the CTR. Vartak et al. (2017) propose a meta-learning strategy to address the cold start problem when new items arrive continuously. In contrast, our proposed SBL algorithm avoids any extra data or different neural network architectures. Instead, we employ the ϵ-greedy random-exploration scheme and shadow bids to feed more data from new ads into the neural networks, which also substantially increases the accuracy of CTR/CVR estimations.

We also study the cold start problem as a data-driven optimization model and offer efficient algorithms to tackle this challenge. Viewing the problem as ad allocation in the repeated-auction setting is in line with another stream of literature in operations management (see, e.g., Caldentey and Vulcano 2007, Balseiro et al. 2014, 2015, Hojjat et al. 2017, Balseiro and Gur 2019). In particular, Balseiro et al. (2014) adopt a dual-based bid-price control policy to study the ad allocation problem in the presence of the trade-off between short-term revenue and long-term value from delivering good spots to the (contracted) reservation ads. While most of the literature on ad allocation assumes the CTR is known to the decision-maker, we study a more realistic contextual bandit setting where the true CTR is unknown and is predicted by an underlying machine learning system.

Our bandit algorithm is closely related to the literature on the stochastic contextual bandit. We compare our algorithm's properties with existing contextual bandit algorithms in Table 1, where the settings consistent with the practice of Platform O are marked in bold. At a high level, contextual bandit algorithms can be categorized into two different classes (see Simchi-Levi and Xu 2020): (1) *agnostic* approaches, which are model-free but require a prespecified policy set and optimization oracles; and (2) *realizability-based* approaches, which explicitly specify the underlying model to represent the reward as a function of contexts. As Simchi-Levi and Xu (2020) observe—"While many different contextual bandit algorithms (realizability-based or agnostic) have been proposed over the past twenty years, most of them suffer from either theoretical or practical issues"—there is still substantial room for improvement in this literature. It is useful to differentiate our algorithm from existing ones with both agnostic and realizability-based approaches.

The agnostic approaches for contextual bandits (e.g., Dudik et al. 2011, Agarwal et al. 2014, Agrawal et al. 2016) usually adopt a conservative exploration scheme (Bietti et al. 2018), based on an AMO and a policy set. Take, for example, the algorithm proposed by Agarwal et al. (2014) with a $\tilde{O}(\sqrt{KT\log(|\Pi|)})$ regret bound. They assume that, given the policy set Π and the set $\mathcal{S}$ of context-reward pairs $(x,r) \in X \times \mathbb{R}^K$, the AMO returns the loss-minimization policy $\pi^* = \arg\max_{\pi\in\Pi} \sum_{(x,r)\in\mathcal{S}} r(\pi(x))$. The reason to assume this oracle as a subroutine is that it is generally

impractical to optimize the loss by enumerating over Π. In a practical setting such as the problem we study, this oracle is clearly infeasible. First, Platform O leverages the deep neural network, a very large policy set in which $|\Pi|$ is on the magnitude of trillions. Specifically, if the policy set Π is the collection of neural networks with fixed structure, depth, and width, even under the proper parameter discretization, its cardinality $|\Pi|$ grows exponentially with the number of parameters. In fact, without a proper "realizability" assumption, the AMO is computationally intractable in practice. Even if we assume the underlying data-generating process is a neural network, we are not aware of an efficient AMO for finding the optimal policy. Second, the algorithm of Agarwal et al. (2014) computes the empirical regret of a policy via *inverse propensity score* (IPS) at each epoch. The IPS method gives an unbiased reward estimate of a policy and is, thus, widely used in regret analysis. However, IPS suffers from a high variance when the policy set is large and/or the past sample paths vary significantly, which is indeed the case of our implementation on Platform O. In fact, all the agnostic approaches in the contextual bandit literature suffer from the aforementioned two issues and are, therefore, not applicable in our context.

The core idea of our algorithm is similar to the realizability-based approaches. Some realizability-based algorithms leverage the Upper Confidence Bound (UCB) and Thompson sampling exploration schemes, which are only tractable for reward functions parametrized in a certain way, such as linear models Chu et al. (2011) and deep neural networks Zhou et al. (2020). Foster et al. (2018) base their realizability-based algorithm on a least-squares regression oracle, which is amenable to widely used gradient-based training methods. Empirically, this algorithm works well among existing contextual bandit approaches, but it is theoretically suboptimal. In the realizability-based setting with an offline regression oracle to predict the reward, Simchi-Levi and Xu (2020) provide the first optimal black-box reduction (i.e., achieving the $\tilde{O}(\sqrt{T})$ theoretical lower bound) from a contextual bandit to an offline regression. The key to this reduction is a special exploration-exploitation scheme implicitly aligned with the agnostic approach proposed by Agarwal et al. (2014). This reduction works only for an objective function linear in the accumulated reward, so it is not applicable to our setting of contextual bandits with a concave objective function. It remains an open problem whether such a reduction that matches the $\tilde{O}(\sqrt{T})$ theoretical lower bound exists for contextual bandits with concave objectives (e.g., Agrawal and Devanur 2014, Agrawal et al. 2016). In this paper, we integrate a machine learning oracle into a contextual-bandit model with a concave objective function, and we develop dual-based algorithms that achieve a sublinear regret. Moreover, implementing these known approaches in the literature requires substantial engineering effort, such as a complex sampling scheme for known policies, whereas our method is compatible with the existing advertising system on Platform O.

Table 1 Algorithm's Performance in the Contextual Bandit Setting

Algorithm	Bandit Setting	Regret	Computational Complexity
LinearUCB (Agrawal and Devanur 2014)	linear context **knapsack**	optimal	calls to offline linear regression at each round
NeuralUCB (Zhou et al. 2020)	**neural network context** non-knapsack	optimal	gradient-descent-based update of the predictor at each round
Regressor Elimination (Agarwal et al. 2012)	**realizability-based** non-knapsack	optimal	$\Omega(\lvert\Pi\rvert)$ intractable
ILOVETOCONBANDITS (Agarwal et al. 2014)	agnostic non-knapsack	optimal	$\tilde{O}(\sqrt{KT/\log\lvert\Pi\rvert})$ calls to AMO
Algorithm adapted from ILOVETOCONBANDITS (Agrawal et al. 2016)	agnostic **knapsack**	optimal	$\tilde{O}(K\sqrt{KT\log\lvert\Pi\rvert})$ calls to AMO
RegCB (Foster et al. 2018)	realizability-based non-knapsack	suboptimal	$O(T^{3/2})$ calls to an offline regression oracle
FALCON (Simchi-Levi and Xu 2020)	realizability-based non-knapsack	optimal	$O(\log T)$ calls to an offline regression oracle
SBL-RS/SBL-DMD (this paper)	**neural network context knapsack**	suboptimal	gradient-descent-based update of the predictor, $O(T^{1/3})$ calls to solving dual or dual mirror descent

Besides the literature on contextual bandit algorithms, our work is also closely related to the growing literature on solving operations management problems with online learning. Given the intrinsically uncertain business environment, a recent trend is to combine learning theory and optimization to solve revenue management and inventory control problems. For example, Chen et al. (2019) build an algorithm to solve the joint problem of pricing and inventory control with nonparametric demand learning for nonperishable products, and they show the regret convergence result. Nambiar et al. (2019) propose an algorithm with theoretical performance guarantees to solve the dynamic pricing problem with misspecified demand models; they evaluate its performance using offline simulations. Ferreira et al. (2018) use Thompson sampling to learn the demand at each price and solve the network revenue management problem. Chen et al. (2020) build an online learning algorithm to solve the single-item inventory-control problem under the periodic review, backlogging policy with unknown capacity and demand distributions. Chen and Gallego (2018) propose a primal-dual learning algorithm to learn the dual optimal solution for the personalized dynamic pricing problem with an inventory constraint. Bastani et al. (2019) propose a metadynamic pricing algorithm to learn the prior through experiment while solving the pricing problem. Golrezaei et al. (2019) propose learning algorithms to set reserve prices in contextual auctions. Our main contribution to this strand of literature is that we have not only proved the theoretical performance guarantee of the proposed algorithm but also implemented it on a large-scale advertising platform and tested its performance using field experiments.

Last but not least, our paper directly relates to the growing literature on field experiments in online platforms (Terwiesch et al. 2020). For example, Zhang et al. (2020) document the spillover effects across platform users in a field experiment on a retailing platform. Zeng et al. (2020) show that social nudge can boost the productivity of content providers on a social network platform

through randomized field experiments. Fisher et al. (2018) leverage both modeling and field experiments to study competition-based dynamic pricing in retailing. Several other papers in the literature conduct field experiments to study platform operations problems (e.g., Cui et al. 2019a, Feldman et al. 2018, Cui et al. 2019b). In the marketing literature, Schwartz et al. (2017) implement a learning algorithm to optimize the user-acquisition strategy through display advertising and conduct a field experiment to gauge its effectiveness. A growing body of literature examines the violation of SUTVA for experiments on large-scale platforms. Ha-Thuc et al. (2020) develop a new counterfactual framework for seller-side A/B testing on Facebook Marketplace and show that the new experiment framework satisfies the SUTVA. Johari et al. (2020) propose a mean-field model to show that single-sided experiments (demand-side or supply-side randomization) will result in biases in estimation for a two-sided marketplace such as Airbnb. They also propose a two-sided randomization and the associated estimator, which is unbiased when the supply and demand are extremely imbalanced. Other authors use clustering algorithms to reduce the impact of interference in experiments on networks. Rolnick et al. (2019) propose a geographical clustering algorithm (referred to as the GeoCUTS algorithm) that minimizes the interference between different geographical units while preserving the balance in cluster size. Pouget-Abadie et al. (2019) introduce a novel clustering objective and a corresponding algorithm that partitions a bipartite graph so as to maximize the statistical power of a bipartite experiment on that graph. Our contribution toward this stream of research is the design and implementation of a novel, two-sided randomized field experiment to causally estimate the value of our proposed bandit algorithm. The proposed experiment framework could potentially be applied to evaluating other algorithms and policies in general recommender systems of large-scale, two-sided online platforms.

3. Background and Model

In this section, we first introduce the background setting of a typical demand-side platform (DSP) for online advertising, with a particular focus on its cold start problem. Based on institutional knowledge, we then develop a data-driven optimization model that integrates linear programs and multi-armed bandits to tackle the issue.

3.1. Online Advertising Platforms

Large-scale online platforms such as Facebook and TikTok are usually equipped with a DSP, a centralized advertising system that aggregates online ads and efficiently matches the ads with users. Figure 2 summarizes the landscape of a DSP. Advertisers and platform users interact with each other on the DSP. On the demand side, advertisers set up their advertising campaigns by submitting the necessary information to the DSP: bid prices, billing options, ad content, advertising budget, and the users they wish to target. On the supply side, platform users are exposed to ads while

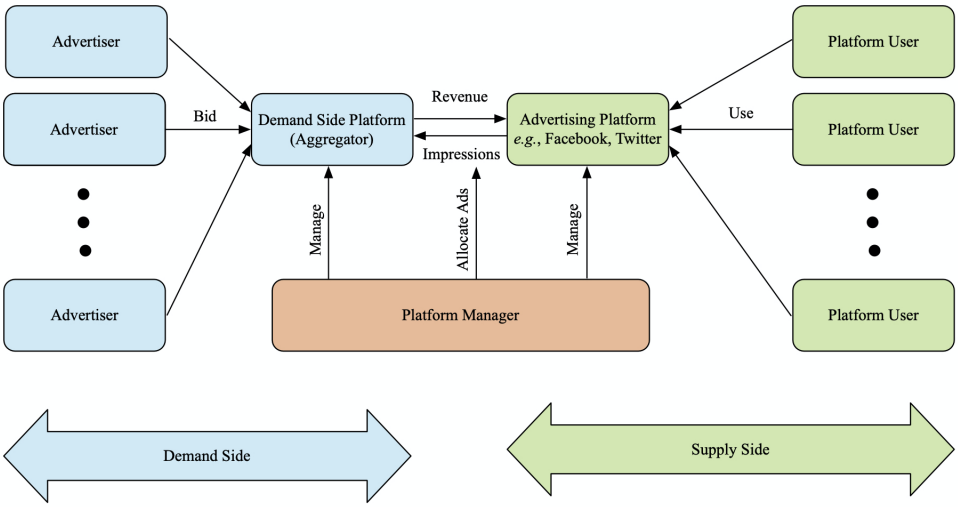


Figure 2 Online Advertising on Platforms

viewing organic content. The DSP plays a central role in allocating user impressions to different ads, with a goal of maximizing long-term revenue.

Next, we show how the DSP monetizes its user traffic. Ad impression requests from platform users continuously arrive at the DSP. For the rest of this paper, we use ad impression, user view, and user impression interchangeably. For large-scale online platforms, the DSP allocates billions of impressions to hundreds of thousands of ads each day—the decision is typically based on a large-scale auction, where hundreds of ads compete to win an ad impression based on advertisers' bids, predicted click-through rates (pCTR), and predicted conversion rates (pCVR). The user impression is allocated to the ad with the highest *estimated Cost per Mille* (eCPM) of the match between the impression and the ad, which measures the expected revenue of displaying the ad to the respective platform user a thousand times. This rule ensures that each ad impression generates the highest ex ante revenue in expectation.

Advertisers can choose from among several billing options, depending on what they wish to bid for each impression (e.g., clicks or conversions) and how the advertising fee is charged (e.g., by impressions, clicks, or conversions). Under all billing options, the bids, pCTRs, and pCVRs can be effectively converted to eCPM, based on which ads can be ranked under the same scale. See Appendix H for a detailed description of different auction mechanisms and billing options.

Cold Start on a DSP. In Section 1, we noted that the inability to accurately predict the CTR and CVR of new ads makes cold start one of the key challenges faced by platforms and advertisers alike. It is extremely difficult to strike a smart balance between boosting new ads that have great potential to enhance the long-term thickness of the platform and maximizing the short-term revenue generated by high-quality mature ads. To the best of our knowledge, most DSPs

tackle the cold start problem ad hoc. For example, to increase its cold start success rate, Platform O has adopted a bid-controlling system (called the PID system; see Appendix H) to *uniformly* increase the system bidding prices for all new ads within a very short time until the preset upper bounds of the system bidding prices are met. This approach increases the probability of winning impressions for new ads, resulting in more exploration of the ads and potentially more conversions. Soon after such sharp increases in the system bidding prices, this system will adaptively lower these prices to offset the extra costs caused by the uniform bid increases. To our knowledge, this heuristic approach has no performance guarantee—fine-tuning hyperparameters is the only leverage. In the following, we formulate the cold start problem as a data-driven optimization problem, design a contextual bandit algorithm with a provable performance guarantee to address this problem, and conduct a two-sided experiment to evaluate the proposed algorithm.

3.2. The Cold Start Model

We formulate the cold start problem of a DSP as a data-driven optimization model. To highlight the key trade-off associated with the problem and avoid unnecessary complexity, we make two high-level modeling assumptions. First, the ad allocation mechanism is a first-price auction, where all advertisers bids on clicks, so they are charged once their ad is clicked. Without loss of generality and for ease of exposition, we assume CVR=pCVR= 1, i.e., conversion is guaranteed upon click-through. (Later, we will show that our proposed algorithm can be easily implemented on a real DSP.) The first-price auction is more intuitive for advertisers, so there is a recent trend of switching from second-price auctions to first-price auctions in the online advertising industry. For example, Google Ad Manager moved to first-price auctions in 2019.[6] Second, the real-time system bidding price of each ad remains the same as the bid submitted by the advertiser. In other words, for our theoretical model, we set $k_p = k_i = k_d = 0$ in the PID system (see Appendix H). In the online implementation of our proposed algorithm, the model is adapted to incorporate the actual online auction mechanism and the real-time system bidding prices of the DSP we experiment on.

We consider a DSP where a set of K new ads, denoted as $A := [K] = \{1, 2, ..., K\}$,[7] are competing for user impressions. We consider only new ads in our base model; in Section 5.2, we discuss how our algorithm can be generalized to a setting with both new and mature ads. User impressions arrive sequentially at the DSP. We define the set of all user impressions as $[T] = \{1, 2, ..., T\}$. For each user impression t and ad j, there is an associated context/feature vector $x_{t,j}$. The context $x_{t,j}$ could be quite broad, containing the demographic and behavioral information of user impression t inherited from the platform, and ad information from ad j. Upon the arrival of user impression

[6] https://www.blog.google/products/admanager/rolling-out-first-price-auctions-google-ad-manager-partners/.

[7] For ease of reading, we list all the notations of the paper in Table 4, Appendix A.

t, the DSP observes K feature vectors $x_{t,j},\ j \in A$. For ease of exposition, we define the vector $x_t := (x_{t,1}, x_{t,2}, \ldots, x_{t,K}) \in X$, where X is a countable feature space. Suppose that ad $a_t \in A$ is chosen to be displayed. We can define a $K-$dimensional binary vector $v_t(a_t) \in \{0,1\}^K$ representing whether each ad is clicked. More specifically, the j-th component of the vector $v_{tj}(a_t) = 1$ only if $a_t = j$ and ad j is clicked by user t. Furthermore, we assume a *stochastic contextual bandit setting*, i.e., the set of context vectors and the click through vector $(x_t, \{v_t(a)\}_{a \in K})$ for $t \in [T]$ is drawn *i.i.d.* (independent and identically distributed) from a distribution $\mathcal{D}$ over $X \times \{0,1\}^{K^2}$, which is unknown to the DSP. And, we can observe the partial outcome $v_t(a_t)$ only at round t of the played ad a_t. Throughout this paper, we use the subscript i to denote the context index of the countable feature space X. We denote the marginal distribution of $\mathcal{D}$ over the context as $\mathcal{D}_X$, i.e., for round $t \in [T]$, the context type i is drawn i.i.d as $i \sim \mathcal{D}_X$. Given the context information for round t and ad j, we define $c_{tj} := \mathbb{E}[v_{tj}(a_t) | a_t = j]$ as the CTR of ad j at round t. We sometimes also abuse the notation, denoting by c_{ij} the CTR of ad j under the context type i.

A core challenge faced by Platform O and other DSPs is jointly optimizing the revenue and the cold start reward of new ads. To evaluate revenue during cold start, we define $V := \sum_{t=1}^{T} v_t(a_t)$ as the accumulative click-through vector, where $V_j := \sum_{t=1}^{T} v_{tj} \in \{0, 1, ..., T\}$ is the total number of clicks generated by ad j until customer T. As prescribed by the oCPC billing option (see Appendix H for details), the total revenue generated by the ads is given by

$$\sum_{j=1}^{K} b_j V_j,$$

where $b_j \in [0,1]$ is the bid (per click) of ad $j \in A$. To quantify the cold start reward, one may want to directly estimate the total lifetime revenue from an ad based on the number of accumulated conversions during its cold start period (the first three days for Platform O). However, such an estimation is extremely difficult, if not impossible, because we need to establish the causal effect of conversions during the cold start period on the new ad's lifetime revenue. Therefore, we take an alternative approach to approximating the aforementioned relationship between conversions in the cold start period and lifetime revenue. We observe from Figure 1 that an ad's retention rate increases linearly in the number of clicks/conversions while this number is below a certain threshold; it stays (almost) unchanged once it exceeds the threshold. Motivated by this phenomenon, we assume the cold start reward of each conversion for ad j before the number of accumulated conversions reaching the conversion target as $\beta_j \in (0,1]$. Without loss of generality, we denote the conversion target as αT, where $\alpha \in (0,1)$. Thus, the cold start reward is given by

$$\sum_{j=1}^{K} \beta_j \min\{V_j, \alpha T\}. \tag{1}$$

In practice, the conversion target αT is determined by business practice and validated by our observation in Figure 1. We specify the cold start reward per conversion β_j via two steps: (1) Inherit the business practice of Platform O that $\beta_j = 2b_j$ for each ad j, and (2) conduct simulations to validate the choice of β. Our simulation results, in Appendix D, demonstrate that setting $\beta_j = 2b_j$ for each ad j would significantly increase the expected long-term revenue for the platform. Furthermore, our two-sided experiment demonstrates that such a choice of β_j boosts the long-term advertising revenue of Platform O by 5.35%.

We are now ready to present the objective of the DSP for cold start, which equals the sum of revenue and cold start reward:

$$\Gamma(V) := \sum_{j=1}^{K} b_j V_j + \sum_{j=1}^{K} \beta_j \min\{V_j, \alpha T\} = \sum_{j=1}^{K} b_j \sum_{t=1}^{T} v_{tj} + \sum_{j=1}^{K} \beta_j \min\left\{\sum_{t=1}^{T} v_{tj}, \alpha T\right\}, \tag{2}$$

which is piecewise linear and concavely increasing in the number of conversions for each ad j.

3.3. Definition of Regret

In this subsection, we formally define the benchmark for our proposed bandit algorithms. In every round $t \in [T]$, a policy π observes the feature vector $x_t \in X$, chooses an ad/action $a_t \in A$, and observes the random outcome whether the ad is clicked. We define the history update to round t as $\mathcal{H}_t = \bigcup_{s=1,\ldots,t-1}\{(x_s, a_s, v_s(a_s))\}$. Let $\Delta_A := \{y \in \mathbb{R}^{|A|} : y_j \geq 0, \forall j \in A, \sum_{j\in A} y_j \leq 1\}$ be the set of the non-negative weight/distribution over arms. Formally, a policy π defines a mapping from the history $\mathcal{H}_t$ and the context x_t to the set of distribution over arms Δ_A for any t.

Recall that one can express the expected reward we gain from policy π as $\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(V)\big]$, where $\mathcal{D}^T$ refers to T independent copies of the distribution $\mathcal{D}$. We notice that $\Gamma(\cdot)$ is concave in V. Using Jensen's Inequality, one can show that the following lemma holds.

Lemma 1. *For any policy π, the scaled expected reward can be upper-bounded as*

$$\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(V)\big] \leq \mathsf{OPT} := \max_{y_i\in\Delta_A,\forall i}\left\{\sum_{j=1}^{K}\mathbb{E}_{i\sim\mathcal{D}_X}\left[c_{ij}y_{ij}b_j\right] + \sum_{j=1}^{K}\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}], \alpha\right\}\right\}.$$

Essentially, OPT is the upper bound of the cold start objective function $\Gamma(\cdot)$; it can be viewed as the solution to a fluid version of our cold start problem, in which the decision variable $y_i \in \Delta_A$ is a sampling distribution over all the ads in A given user context i. Similar upper bounds are widely used in the revenue management literature (Gallego and Van Ryzin 1994, Golrezaei et al. 2014, Zhang et al. 2018), as well as the bandit learning literature (Badanidiyuru et al. 2013, Agrawal et al. 2016). With Lemma 1, one can formally define the regret for an arbitrary policy π as

$$\mathsf{Reg}(\pi) = T\cdot\mathsf{OPT} - \mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(V)\big]. \tag{3}$$

Our goal is to propose a novel policy that has a provable performance guarantee (measured by a sublinear regret) and that can be effectively implemented on a practical DSP. As mentioned in the literature review, although the regret defined similar to (3) is common in the stochastic bandit setting as a concave objective function, (see, e.g., Agarwal et al. 2014, Agrawal et al. 2016), the existing bandit algorithms in this literature are not practically feasible in our setting, for two reasons. First, these algorithms are often built upon the AMO, which is computationally intractable for most policy classes. Furthermore, a practical DSP often relies on a machine learning system to generalize the knowledge learned from the data on observed click-throughs and conversions and make accurate predictions about future user behaviors. How to design efficient algorithms for realizability-based bandits with deep learning models and concave objectives remain an open question in the literature. Second, existing MAB algorithms usually provide an empirical estimate of OPT based on IPS (e.g., Agrawal et al. 2016). In practice, such IPS technique may suffer from a high variance. As such, we design a novel primal-dual-based algorithm by leveraging the predictions of the machine learning system of a DSP as model inputs and adding "shadow bids" to new ads. The ad allocation policy can be easily implemented by the auction system of a real-world DSP, by adjusting the bidding prices of new ads. As we show in Section 5, one can easily implement this "shadow bidding with learning" algorithm on a real-world DSP, and our field experiments show significant improvements in long-term retention and advertising revenue without much compromising short-term revenue.

4. Cold Start Algorithms

In this section, we propose novel bandit learning algorithms for our ad allocation model with a cold start reward. Our algorithms leverage the ϵ-greedy exploration strategy, the prediction power of a DSP's underlying machine learning system, and the empirically optimal dual solution to the fluid upper bound.

4.1. Shadow Bidding with Learning (SBL) Algorithms

In this subsection, we outline our primal-dual-based learning algorithm. In designing the algorithm, one central difficulty is the unknown distributional information of the underlying model. In particular, the CTRs at round t are unknown to any online algorithm. Instead, we have access to an empirically estimated CTR only via the online training of predicting models on the historical data. To get an empirically optimal ad allocation policy, one can solve the following ad allocation model at round t:

$$\max_{y_i \in \Delta_A, \forall i \in \mathcal{I}} \sum_{i \in \mathcal{I}} \sum_{j \in A} \hat{p}_i^t \hat{c}_{ij}^t b_j y_{ij} + \sum_{j \in A} \beta_j \min \left\{ \sum_{i \in \mathcal{I}} \hat{p}_i^t \hat{c}_{ij}^t y_{ij}, \alpha \right\}. \tag{4}$$

We assume the number of contexts is countable, and we define $\mathcal{I} := \{1, 2, 3, \dots\}$ as the set of context types. We denote p_i as the probability that incoming context is type i. In our implementation and

analysis, we relax this assumption by considering that p_i is unknown in prior and is estimated using the empirical estimation $\hat{p}_i^t$ based on the historical data $\mathcal{H}_t$. Here, the empirical CTR $\hat{c}_{ij}^t$ is the estimated CTR at time t under the context i, and ad j trained on the historical data $\mathcal{H}_t$. In practice, $\hat{c}_{ij}^t$ is usually produced by a deep neural network associated with the DSP to predict the CTR/CVR of the ads facing different user contexts. By introducing an additional variable u_j for each ad j, we can transform (4) to a linear program:

$$\begin{aligned} \max_{y,u\geq 0} \quad & \sum_{i\in\mathcal{I}}\sum_{j\in A}\hat{p}_i^t\hat{c}_{ij}^t b_j y_{ij} + \sum_{j\in A}\beta_j(\alpha - u_j) \\ \text{s.t.} \quad & \sum_{j\in A} y_{ij} \leq 1,\ \forall i\in\mathcal{I},\ \sum_{i\in\mathcal{I}}\hat{p}_i^t\hat{c}_{ij}^t y_{ij} + u_j \geq \alpha,\ \forall j\in A \end{aligned} \tag{5}$$

We succinctly write the dual of (5) as

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{I}}\hat{p}_i^t \max_{j\in A}\left\{\hat{c}_{ij}^t(b_j+\lambda_j)\right\} + \alpha\sum_{j\in A}(\beta_j - \lambda_j) \\ \text{s.t.} \quad & 0\leq\lambda_j\leq\beta_j,\ \forall j\in A, \end{aligned} \tag{6}$$

which is a nonsmooth convex program with decision variables $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_K)$. Strong duality dictates that the optimal values of (5) and (6) must be the same. Utilizing such duality, we propose the following cold start algorithm.

Shadow Bidding with Learning and Re-Solving (SBL-RS)

Parameters: Epoch schedule $1 = \tau_1 < \tau_2 < \dots$ such that $\tau_m - \tau_{m-1} = \tau_{m+1} - \tau_m \leq O(T^{\frac{2}{3}})$. Cold start reward coefficient β. Target conversion parameter α.

Initialization: $\lambda^1 = 0$, $t = 1$, $m = 1$

For $t = 1, 2, \dots, T$ **do**

Step 1: Observe the context i_t at period t. With probability $\epsilon_t = t^{-\frac{1}{3}}(K\log t)^{\frac{1}{3}}$, the algorithm picks an ad uniformly at random. Otherwise, display an ad $a_t \in \arg\max_j \hat{c}_{i_t j}^t(b_j + \lambda_j^{\tau_m})$ with arbitrary tie-breaking rules.

Step 2: If $t = \tau_m$, we solve the dual model (6) to update λ^{τ_m} by the subgradient descent algorithm and set $m = m + 1$.

Step 3: Observe the outcome of a_t, and update the parameters of the underlying machine learning model for predicting $\hat{c}_{ij}^{t+1}$.

Several remarks are in order. First, we highlight a compelling advantage of the SBL-RS algorithm: it fits perfectly into the auction system of a real-life advertising platform, fully leveraging the predictive power of the embedded machine learning oracle to estimate the CTR of ads. This makes our algorithm generalizable and implementable for any large-scale DSP. In particular, we periodically re-solve the optimization problem (6) to produce the dual vector λ^{τ_m}. With the most recent λ^{τ_m}, and the most up-to-date CTR estimation given the context, $\hat{c}_{ij}^t$, the SBL-RS algorithm picks ad j with the highest adjusted eCPM, $\hat{c}_{ij}^t(b_j + \lambda_j^{\tau_m})$ (ties broken arbitrarily), which can be

easily implemented in practice by adding $\lambda_j^{\tau_m}$ to the bidding price of ad j in the auction system (see the oSBL algorithm presented in Section 5). Solving the dual problem with a carefully chosen epoch will save computing resources without hurting algorithm performance.

Second, the term *shadow bidding* comes from the ad-selection rule. We pick the ad with the largest adjusted eCPM upon the arrival of each user, which is the sum of the bid price, b_j, and the shadow price for the cold start reward, $\lambda_j^{\tau_m}$, multiplied by the predicted CTR. The original bidding process of the DSP seeks only to maximize the short-term revenue by picking ad j with the highest eCPM$= \hat{c}_{ij}^t b_j$, whereas we add $\lambda_j^{\tau_m}$, the shadow price associated with the constraint $\sum_{i\in\mathcal{I}} \hat{p}_i^t \hat{c}_{ij}^t y_{ij} + u_j \geq \alpha$ in (5), to the bid b_j in order to capture the long-term cold start reward of displaying ad j immediately. In effect, we use the solution in the dual space to characterize the correct assignment in the primal space, which gives a fast, simple allocation rule. Similar dual-based strategies are used in the online linear program under stochastic input or random permutation (Li et al. 2020), and the noncontextual knapsack bandit setting (Badanidiyuru et al. 2013). One may also wonder whether existing algorithms that directly solve the primal problem (e.g., Agarwal et al. 2014, Agrawal et al. 2016) would also work in our cold start setting. In fact, though theoretically possible, directly implementing the primal solutions on a practical DSP is very hard, if not impossible. Specifically, solving the primal problem amounts to dictating an ad-assignment scheme. However, the primal space of the problem is extremely large—its solution has a cardinality of the number of impressions multiplied by the number of ads, which is on the order of trillions. The cardinality of the associated dual space, however, is the number of ads, which is on the order of hundreds of thousands. Therefore, in practice, working with the dual space is substantially simpler than working with the primal space.

Third, in each round t, we explore new ads with probability $t^{-\frac{1}{3}}\left(K\log t\right)^{\frac{1}{3}}$, and exploit, with probability $1-t^{-\frac{1}{3}}\left(K\log t\right)^{\frac{1}{3}}$, following the dual-based policy (6). This exploration-exploitation schedule is common for ϵ-greedy algorithms in the bandit learning literature. The novelty of our algorithm lies in the well-designed, dual-based exploitation scheme and the integration of an MAB algorithm with the underlying machine learning oracle of a DSP. One may also consider other exploration-exploitation strategies such as *Upper Confidence Bound* or *Thompson Sampling*. We leave it to future researchers to study the optimal exploration scheme for our cold start problem. Moreover, an ϵ-greedy based algorithm (such as SBL-RS) can be naturally embedded into a DSP in practice without much engineering change, as we will show in Section 5.

Although the SBL-RS algorithm achieves a sublinear regret bound and is therefore asymptotically optimal (see Theorem 1 below), it needs to solve the dual program (6) for at least $O(T^{\frac{1}{3}})$ times, which may be computationally intractable. Therefore, to reduce the computational burden, we incorporate the dual mirror descent method (e.g., Balseiro et al. 2020) into our SBL algorithmic

framework so that it suffices to update the shadow bids by dual mirror descent without solving the dual program throughout the algorithm. Specifically, let $\varphi(\cdot)$ be a $\sigma-$strongly convex function with respect to the L_1-norm (i.e., $\varphi(\lambda) \geq \varphi(\lambda_0) + \langle \nabla\varphi(\lambda_0), \lambda - \lambda_0\rangle + \frac{\sigma}{2}||\lambda - \lambda_0||_1^2$ for any λ and λ_0, where $\nabla\varphi(\cdot)$ is the gradient of $\varphi(\cdot)$) and define the Bregman divergence associated with $\varphi(\cdot)$ as:

$$D_\varphi(\lambda_1, \lambda_2) := \varphi(\lambda_1) - \varphi(\lambda_2) - \langle \nabla\varphi(\lambda_2), \lambda_1 - \lambda_2\rangle. \tag{7}$$

We are now ready to present the SBL algorithm incorporated with dual mirror descent (SBL-DMD) as follows.

Shadow Bidding with Learning and Dual Mirror Descent (SBL-DMD)

Parameters: Cold start reward coefficient β, target conversion parameter α, and step-size η.

Initialization: $\lambda^1 = (0, 0, ..., 0) \in \mathbb{R}^K$.

For $t = 1, 2, \ldots, T$ **do**

Step 1: Observes the context i_t at period t. With probability $\epsilon_t = t^{-\frac{1}{3}}\left(K \log t\right)^{\frac{1}{3}}$, the algorithm picks an ad uniformly at random. Otherwise, display an ad $a_t \in \arg\max_j \hat{c}^t_{i_t j}(b_j + \lambda^t_j)$ with arbitrary tie-breaking rules.

Step 2: Updating λ via the online dual mirror descent. Let $s_t(\lambda) = -\sum_{j\in[K]\setminus a_t} \alpha\lambda_j + (\hat{c}^t_{i_t a_t} - \alpha)\lambda_{a_t}$. Let $z_t \in \partial_\lambda s_t(\lambda)$ be a subgradient and

$$\lambda^{t+1} = \underset{0\leq\lambda_j\leq\beta_j, \forall j\in A}{\arg\min} \langle z_t, \lambda\rangle + \frac{1}{\eta} D_\varphi(\lambda, \lambda^t). \tag{8}$$

Step 3: Observe the outcome of a_t, and update the parameters of the underlying machine learning model for predicting $\hat{c}^{t+1}_{ij}$.

Incorporating dual mirror descent into our SBL framework frees our algorithm from solving the empirical dual (6). Instead, the dual variables are updated with the help of the Bergman divergence $D_\varphi(\cdot,\cdot)$ via a convex optimization (8). In particular, if the strongly convex function $\varphi(\cdot)$ is properly chosen, the dual variable update (8) has a closed-form solution and, thus, can be obtained very efficiently. We show in Theorem 1 that, by setting the learning rate to $\eta = \Theta(1/\sqrt{T})$, the regret of SBL-DMD is of the same order as SBL-RS. In our Platform O implementation, we adapt SBL-RS to the practical online advertising system (the oSBL algorithm presented in Section 5).

4.2. Analysis of the Regret Bound

Notice that in running the algorithm, we are effectively solving

$$\mathsf{OPT}^t = \min_{0\leq\lambda_j\leq\beta_j, \forall j\in A} \sum_{i\in\mathcal{I}} \hat{p}^t_i \max_{j=1,2,\ldots,K} \left(\hat{c}^t_{ij}(b_j + \lambda_j)\right) + \alpha\sum_{j=1}^K (\beta_j - \lambda_j), \tag{9}$$

where $\hat{c}^t_{ij}$ is the estimate of c_{ij} produced by the underlying prediction model prior to round t, and $\hat{p}^t_i$ denotes the empirical distribution of contexts prior at round t. Before formally presenting the results of our main regret analysis, we address two basic issues regrading this formulation. First, we need to bound the gap between optimal empirical *primal* allocation and our optimal empirical

dual allocation. By strong duality, this gap is induced by tie-breaking in Steps 1 and 2 of the SBL algorithms. As we will show in Appendix B, adding an arbitrarily small perturbation to the CTR estimate $\hat{c}_{ij}^t$ will ensure that the tie-breaking in Step 1 will only induce an arbitrarily small additional regret. To bound the gap from tie-breaking in Step 2, we make the following assumption.

ASSUMPTION 1. *For each context $i \in \mathcal{I}$, each ad $j \in A$ and each period t, it holds that*

$$\hat{p}_i^t \hat{c}_{ij}^t \leq O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{-\frac{5}{3}}).$$

Assumption 1 states that the empirically estimated probability of a user with context i clicking ad j is negligible. This assumption is introduced to guarantee that the error from tie-breaking in Step 1 of the SBL algorithms is small. Similar assumptions are made for other primal-dual settings (e.g., Devanur and Hayes 2009, Agrawal et al. 2014). We note that a typical online DSP faces hundreds of millions of different users, each of whom can be regarded as a unique context. Therefore, Assumption 1 is made without loss of generality in practice. We remark that the subgradient descent approach in Step 2 of SBL-RS may also incur some computational errors. To demonstrate the effectiveness of the subgradient descent algorithm in our ad allocation model, in Appendix E, we report a numerical experiment which shows that the solution of our primal-dual subgradient descent process produces almost the same objective function value as the exact solution in the primal space by the Simplex method.

The second issue of the SBL algorithms is the prediction error associated with the underlying machine learning model to estimate CTR. Clearly, the performance of our algorithms depends on that of the underlying predictor for estimating CTR. In practice, the underlying predictor returns the predicted CTR in period t, $\hat{c}_{ij}^t$, by training from a class of functions $\mathcal{X}$ that estimates the CTR of each context facing each ad ($X \times A \mapsto [0,1]$). The CTR predictor may take the form of linear regressors, regression trees, and neural networks, the last of which are the actual case with Platform O. To bound the prediction error of the underlying machine learning model, we make the following Prediction Oracle assumption.

ASSUMPTION 2 **(Prediction Oracle)**. *For each ad $j \in A$ with n_j^t observed i.i.d. contexts drawn from the distribution $\mathcal{D}_X$ before round t and the corresponding click-through outcomes of showing ad j to these contexts, with probability at least $1-\delta$, for any context i, the estimate $\hat{c}_{ij}^t$ satisfies $|\hat{c}_{ij}^t - c_{ij}| \leq O\left(\sqrt{\log(1/\delta) d / n_j^t}\right)$, where d is the effective dimension of the predictor.*

The Prediction Oracle assumption is regardless of the total number of contexts $m = |\mathcal{I}|$. Instead, it assumes that as long as ad j is displayed for a total of n_j^t times with the user contexts drawn in an i.i.d. fashion from the distribution $\mathcal{D}_X$, the error of estimating its CTR has an order of $O\left(\sqrt{1/n_j^t}\right)$ with a high probability, regardless of which contexts the ad is displayed to at round t.

An alternative interpretation of Assumption 2 is that the platform has a machine learning oracle for predicting the ad CTRs with reasonable generalization error, i.e., it is capable of learning from training data and makes accurate predictions on unseen data. In particular, the error decreases with the training sample size n_j^t. Furthermore, the generalization error is negatively impacted by the dimensionality (i.e., complexity) of the underlying problem, which is specified in the assumption by the effective dimension d.

We emphasize that Assumption 2 could be satisfied for general prediction models such as linear regression, regression trees, and neural networks. We first note that, in the noncontextual setting (i.e., X is a singleton), Assumption 2 is reduced to the standard Hoeffding's inequality with the effective dimension $d=1$. If $\mathcal{X}$ is the set of linear regressors and the true data generating process is indeed a linear function, the Prediction Oracle assumption holds with the ridge regression and the effective dimension d defined as the context dimension (Hsu et al. 2014). For the standard ridge regression model, it requires $n_j^t \geq \Omega(d\log(d/\delta))$ in general to achieve this $O\left(\sqrt{1/n_j^t}\right)$ error bound. In our context, as long as we additionally assume $T > Kd$, extra exploration of $O(Kd\log(d/\delta))$ periods before the start of the SBL still suffices to achieve the same regret bound in Theorem 1. Since we are most interested in the dependence of regret on T when T is sufficiently large, this condition is naturally satisfied in this regime. Also notice that in the literature, the dependence of error bound on the effective dimension is either d or $\sqrt{d}$, depending on the regularity conditions of the features (Chu et al. 2011, Yang and Wang 2020, Zhou et al. 2020). We explicitly specify those conditions for neural networks in Appendix G. For the regression-tree predictor, it has been well established in the literature (e.g., Wager and Walther 2015) that an adaptive regression tree with each child node containing at least η fraction of the data points in its parent node and each leaf node containing q training samples has a more general convergence rate of $\sqrt{\frac{\log(n_j^t)\log(d)}{q\log((1-\eta)^{-1})}}$. Therefore, Assumption 2 is satisfied under the mild condition that the regression trees have a fixed depth and $q=\Omega(n_j^t)$, which commonly holds in practice. For a large-scale practical DSP such as Platform O, $\mathcal{X}$ is the set of fully connected neural networks with the ReLU activation function. Assumption 2 holds in this setting with proper parameterization. We defer a detailed discussion of this case to Appendix G. In reality, the number of contexts is large, but thanks to the enormous wealth of data the platform can access, advanced neural network algorithms can extract useful information with small generalization errors. We are now ready to state our main theoretical result in the following theorem.

THEOREM 1 (**$\tilde{O}(T^{\frac{2}{3}})$ Regret Bound**). *Suppose Assumptions 1 and 2 hold, then the we have the following statements.*

(a) The expected regret of the SBL-RS algorithm is bounded by $O(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}})$.

(b) The expected regret of the SBL-DMD algorithm is bounded by $O(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}}) + \mathbb{E}\big[\sum_{t=1}^{T} s_t(\lambda) + \frac{2\eta}{\sigma}T + \frac{1}{\eta}D_\varphi(\lambda,\lambda^1)\big]$. *Thus, taking* $\lambda = 0$ *and* $\eta = \sqrt{\frac{\sigma\bar{D}}{2T}}$ *where* $\bar{D} := D_\varphi(\lambda,\lambda^1)$, *we have the expected regret of the SBL-DMD algorithm is bounded by* $O(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}})$.

Theorem 1 shows that our proposed SBL-RS and SBL-DMD algorithms both have an expected regret of order $\tilde{O}(T^{\frac{2}{3}})$, which is consistent with the ϵ-greedy type algorithms for contextual bandits. Furthermore, the bound depends on the effective dimension of the predictor by $\sqrt{d}$. This is a natural and necessary price we have to pay with the SBL algorithms, which relies on the underlying machine learning model to predict CTR. If the underlying CTR prediction is easy (hard) so that the effective dimension of the predictor is small (large), our algorithm can achieve a sharper (looser) regret bound. Theorem 1 presents our regret bound in the expected regret, but we can easily extend it to a high-probability-type bound using the Azuma-Hoeffding inequality. We also remark that the analysis of our ϵ-greedy based SBL algorithms is more difficult than the standard contextual bandit algorithms with a linear reward function (e.g., Chu et al. 2011). This is because, in our problem, the cold start reward depends on the aggregated click-though outcomes over T periods.

To prove Theorem 1, one has to carefully map the total reward collected so far into the dual space. Specifically, we first establish, in Lemma 2 (in Appendix B), the approximate complementary slackness and bound the duality gap between the empirical primal and the empirical dual due to tie-breaking in Step 1 of the SBL algorithms by $O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})$. Then, we build an auxiliary reward process independent of history: Each click of ad j generates a reward of $b_j + \beta_j$, regardless of whether the threshold αT is met. Based on the approximate complementary slackness and Hoeffding's inequality, Lemma 3 (in Appendix B) bounds, under the SBL-RS algorithm, the gap between the auxiliary reward process and the optimal reward by $O(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}})$. Finally, in Lemma 4 (in Appendix B), we bound the gap between the auxiliary reward process and the true reward process by $O(\sqrt{KT\log T})$. Putting these bounds together would prove the desired regret bound of order $O(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}})$ for the SBL-RS algorithm. For the SBL-DMD algorithm, a key property of dual mirror descent (Proposition 1 in Appendix B.4) implies that, compared with SBL-RS, it incurs only an additional regret of a lower order, $O(\sqrt{T})$. Therefore, the SBL-DMD algorithm also has a regret bound of $\tilde{O}(T^{\frac{2}{3}})$. We refer interested readers to Appendix B for the proof details.

5. Field Experiment Design and Algorithm Implementation

To demonstrate the practical value of our SBL algorithm, we conduct a two-sided randomized field experiment to causally evaluate the impact of the algorithm on both the revenue and the cold start reward/success rate of the DSP. In this section, we first discuss our field setting and introduce our two-sided experiment design, then we illustrate the online implementation of our algorithm (i.e., the oSBL algorithm).

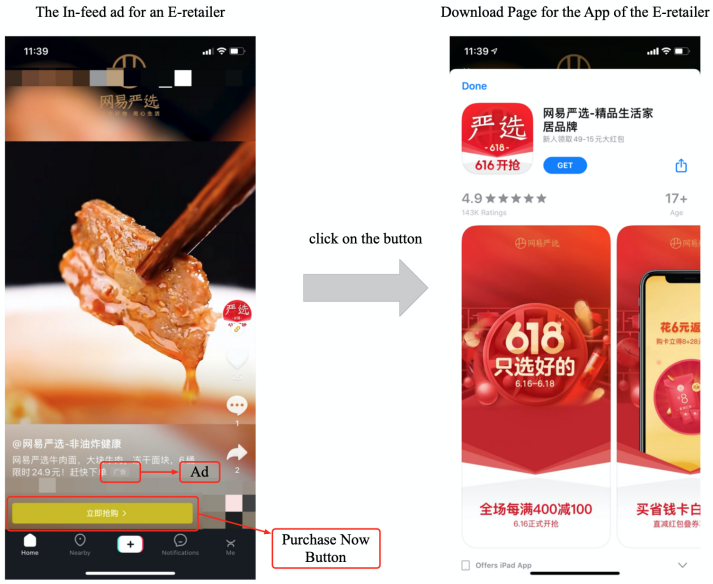


Figure 3 How ads are displayed to users [8]

5.1. Two-Sided Experiment Design

We collaborate with a large-scale online video-sharing platform (Platform O), where *in-feed* advertising contributes to a substantial share of its revenue. Platform O features interactive short videos (whose length is typically no more than 30 seconds) and in-feed ads—more akin to TikTok than to YouTube. On Platform O and other social media platforms, in-feed ads are presented in a short-wide format (with an "Ad" label) and intertwined with other organic content updates. As a user swipes up on the screen, a new organic video or an in-feed ad will be shown. Unlike on YouTube, where users have to watch a certain length of an ad before skipping it, Platform O users can swipe up to skip an ad at any time. Users interested in an ad may click the button that directs them to external sites such as AppStore to download the smartphone app, an e-commerce website for online shopping, and so on. Users are converted if they finish the target action set by the advertiser, such as downloading the app or purchasing the product. Figure 3 illustrates this process.

One may want to test the effectiveness of our algorithm by either randomly assigning new ads or randomly assigning user views into treatment and control groups, as shown in Figure 4 panels (a) and (b), respectively. However, both designs would violate the Stable Unit Treatment Value Assumption (SUTVA, see Imbens and Rubin 2015), thus causing biased estimates for the effect of the new algorithm (Blake and Coey 2014, Johari et al. 2020). Figure 4(a) illustrates the ad-side randomization design, in which new ads are randomly assigned to treatment and control

[8] To protect the Platform O's identity, we created the screenshots of an in-feed ad on another large online advertising platform in Figure 3, whose interface is similar to Platform O.

groups. The SBL algorithm is applied to all ads in the treatment group, and the baseline cold start algorithm of the DSP (the real-time bidding prices generated by the PID controller; see Section 3.1 and Appendix H) is applied to ads in the control group. In this ad-randomization setting, the ads using our new algorithm will *compete* with those using the baseline algorithm on the same set of impressions, so the global effect of the SBL Algorithm (i.e., the effect of the algorithm applied to all the ads on the platform) will be *overestimated* due to the cannibalization effect. This bias has been confirmed by our numerical simulations, which show that the ad-side experiment overestimates the cold start success rate by as much as 120% (see Table 8 in Appendix D.1).

Alternatively, one may conduct an experiment that randomizes over user-views (UVs), in which users are randomly assigned to treatment and control groups each using different algorithms (SBL for treatment and the baseline algorithm for control). See Figure 4(b) for an illustration. Such UV-side randomization design has been widely applied in other online platform contexts (see, e.g., Schwartz et al. 2017). For our setting, however, the UV-side randomization design is also invalidated, again due to the violation of SUTVA. Both the SBL algorithm and the baseline algorithm are applied to the same new ads, through which the effect of SBL will *spill over* to the control group. Specifically, under this experiment design, the SBL algorithm is applied to all new ads so that the underlying machine learning model could produce better CTR estimates for all new ads, which are also served by the baseline algorithm. Therefore, the effect of the baseline algorithm will be overestimated. Due to such spillover effect, directly comparing the outcomes of the treatment- and control-group users under the UV-side randomization will result in *underestimates* for the effect of our algorithm. Our simulation studies have confirmed that such underestimation bias under UV-side randomization could be as high as 40% (see Table 8 in Appendix D.1).

	Treatment New Ads	Control New Ads	Non-Experiment New Ads	Mature Ads
100% UV	Treatment Condition	Control Condition		

(a) Experiment with Ad-Side Randomization

	100% New Ads	Mature Ads
Treatment UV	Treatment Condition	
Control UV	Control Condition	
Non-Experiment UV		

(b) Experiment with UV-Side Randomization

Figure 4 One-Sided Randomization Experiments

To address the aforementioned SUTVA violation issue under one-sided experiments, we design a novel two-sided field experiment to evaluate our SBL algorithm. A similar two-sided experimental

	20% Treatment New Ads	20% Control New Ads	60% Non-Experiment New Ads	Mature Ads
33% Treatment UV	B11	B12	B13	B14
33% Control UV	B21	B22	B23	B24
33% Non-Experiment UV	B31	B32	B33	B34

Figure 5 Two-Sided Experiment Design

Notes: The new ads in cell B11 are bid with the SBL algorithm (i.e., the shadow bids λ^* will be added to the real-time bidding prices). The new ads in cells B21, B31, B12, and B32 are forbidden to join the auction. All other ads in uncolored cells join the auction following the real-time bidding prices without shadow bids.

framework has also been studied by Johari et al. (2020) from a theoretical perspective. Our experiment was conducted from May 23, 2020, to May 30, 2020; the experiment design is illustrated in Figure 5. Specifically, we randomly assigned 33% platform UVs into the treatment group and another 33% UVs into the control group. On the ad side, we randomly assigned 20% of the new ads to the treatment group and 20% to the control group. The rest of the UVs and ads are referred to as the nonexperiment UVs and nonexperiment ads. The ad-side randomization is independent from the UV-side randomization. The SBL algorithm is applied if both UV and ad are in the treatment group (cell B11 in Figure 5), whereas the baseline algorithm is applied if both UV and ad are in the control group (cell B22 in Figure 5). The salient feature of this design is that the treatment (control) ads can bid only on the treatment (control) UVs; they are not allowed to bid on the control (treatment) UVs. Implementation-wise, we set the bids in cells B12, B21, B31, and B32 in Figure 5 to 0. For the nonexperiment new ads, we applied the baseline cold start algorithms regardless of UVs (cells B13, B23, and B33 in Figure 5). Finally, we keep the bidding algorithm for the mature ads (cells B14, B24, and B34 in Figure 5) unchanged.

Through such a two-sided randomization design, the SUTVA condition is restored to the greatest extent we can. First, our two-sided design avoids letting the treatment and control ads directly compete with each other on the same UVs and therefore removes the cannibalization effect of experimentation with ad-side randomization only. Furthermore, blocking the control UV impressions for treatment ads and the treatment UV impressions for control ads (B21 and B12 in Figure 5, respectively) confines the treatment ads to our SBL algorithm and the control ads to the baseline algorithm, thus removing the spillover effect of the experiment only with UV-side randomization. Specifically, the CTR estimates, produced by the underlying machine learning system, for the treated ads will be affected *only* by our SBL algorithm, whereas those for the control ads *only* by the baseline algorithm. Thus, the difference between the treated ads and control ads shall be causally attributed to the effect of our new algorithm compared with the baseline one. To fully ensure SUTVA for our two-sided experiment, we make two additional assumptions during our experiment period: (1) The CTR and CVR distributions of the mature ads are not affected by the

cold start algorithms applied to different UVs; (2) The number of ad impressions displayed to a user is not affected by the cold start algorithms applied to different ads. We report the verification of both assumptions in Appendix C.1.

Some other recent developments on experiment design and analysis have also addressed the violation of SUTVA in a two-sided setting (e.g., Rolnick et al. 2019, Pouget-Abadie et al. 2019). This line of research focuses on developing cluster-level randomization and the associated algorithms to improve the power of statistical inferences in this setting. We, however, take a different approach, proposing a new two-sided experimental framework that causally evaluates an advertising algorithm for a large-scale DSP.

5.2. Online Implementation of the Algorithm

We highlight a key advantage of our SBL algorithms—they can be easily adapted into the infrastructure of Platform O's DSP. Such convenience has enabled us to actually implement the algorithm online. The implemented version of the algorithm is Online Shadow Bidding with Learning (oSBL), as detailed below.

Online Shadow Bidding with Learning (oSBL) Algorithm

Parameters: Set epoch schedule $1=\tau_1<\tau_2<\cdots<\tau_m=T$ with fixed, one-hour intervals; the cold start reward coefficient $\beta_j=2b_j$; and the conversion target $\alpha T=10$ for new ads.

Initialization: $\lambda^1=0$, $t=1$, $m=1$

For $t=1,2,\ldots,T$ **do**

Step 1: Observe the context i_t at round t. Choose the top 150 ads (including new and mature ads, ranked by a preranking model[9]), together with 15 randomly picked new ads, to join the auction.

Step 2: Get $\hat{c}^t_{i_tj}$, the estimate of pCTR×pCVR. Display the ad $a^*_t=\arg\max_{j\in[K_t]}\hat{c}^t_{i_tj}(b^t_j+\lambda^{\tau_m}_j)$, where b^t_j is the system bidding price calculated by real-time PID system for ad j at period t, and $[K_t]=[K^n_t]\cup[K^m_t]$ is the set of 165 ads that join the auction at time t, with $[K^n_t]$ ($[K^m_t]$) as the set of new (mature) ads.

Step 3: If $t=\tau_m$, construct the history data set $\mathcal{H}_t$ by randomly sampling 4% of the auctions in the past hour, the auction/round index set of which is denoted by $\mathcal{T}^t$. Update $m=m+1$, and $\lambda^{\tau_m}_j$ for each ad j by solving the following dual program, where the shadow bidding price for mature ads is set as 0, i.e, $\lambda_j=0,\ \forall j\in[K^m_\tau]$,

$$\begin{aligned}\min\ &\sum_{\tau\in\mathcal{T}^t}\max_{j\in[K_\tau]}\left\{\hat{c}^\tau_{i_\tau j}(\lambda_j+b^s_j)\right\}+\alpha|\mathcal{T}^t|\sum_{j\in[K_\tau]}(\beta_j-\lambda_j)\\ \text{s.t.}\quad&\lambda_j\in[0,\beta_j],\ \forall j\in[K^n_\tau],\ \lambda_j=0,\ \forall j\in[K^m_\tau]\end{aligned}$$

Step 4: Observe the outcome of ad a^*_t, and update the parameters in the neural networks. The advertiser will be charged based on the real-time system bidding price b^t_j, instead of the total bid $b^t_j+\lambda^{\tau_m}_j$.

Several aspects of the oSBL algorithm are different from the original SBL-RS and SBL-DMD. First, oSBL accounts for both new and mature ads, with the shadow bids for mature ads fixed

[9] On Platform O's DSP, there are two stages before an ad enters the final auction: filtering and preranking, both of which adopt deep neural network models to rule out the ads not suitable for the user impression.

at 0. Second, the bid for each ad j in oSBL will follow the system bidding prices generated by the PID system (so the bid will change over time), to which the shadow bids are adaptively added. Furthermore, the two-sided experiments (Figure 5) can also be easily implemented online by adjusting the shadow bids according to which cell the ad-UV pair belongs to. Third, the exploration of the oSBL algorithm is to randomly add 15 new ads into the final auction for each impression, instead of the ϵ-greedy scheme proposed in SBL-RS and SBL-DMD. This adjustment is mainly driven by the fact that we inherit the 10%-exploration heuristic that has already been implemented by Platform O's DSP. Making minimum changes to the online system of the DSP will ensure the robustness of our new algorithm. Fourth, when computing the shadow bids λ_j^* for each ad j, we sample 4% of the total auctions for user impressions. Such a downsampling approach could further reduce the computational burden of the oSBL algorithm. As a matter of fact, our algorithm could produce robust shadow bids even with a sampling rate of only 1%, as shown by our robustness check results in Appendix F.

We also set the fixed, one-hour epoch schedule interval in oSBL to update the shadow bids. On the one hand, this makes the pace of the algorithm consistent with other online systems of the DPS, such as the predictive models for pCTR and pCVR, and the PID controller. On the other hand, it alleviates the computational burden of the algorithm so that the shadow bids can be generated in a timely manner. Also, due to the engineering constraint, the pCTR and pCVR data cannot be accessed in real time by our field experiment in our actual implementation on Platform O—directly re-solving the optimal dual variables for the empirical allocation problem will be more robust than updating via the mirror descent procedure. This is the reason why oSBL is implemented in a re-solving fashion on Platform O.

Finally, we set the cold start reward coefficient $\beta_j = 2b_j$ and the conversion target $\alpha T = 10$ mainly because of the business practice of Platform O's DSP. Furthermore, as shown by our simulation results in Section 6.3 with $\beta_j = 2b_j$, the oSBL algorithm would yield a substantial (at least 5.35%) increase in Platform O's long-term total advertising revenue. The more sophisticated choice of the cold start reward coefficient, therefore, will boost the long-term revenue even higher.

6. Field Experiment Results

In this section, we present the results of our two-sided field experiment. The randomization check (see Appendix C.2 confirms that both the treatment ads and the control ads in our sample are comparable, implying that any difference between groups after the experiment started should be attributed to whether our new oSBL algorithm has been implemented. In the following subsections, we document three sets of results to demonstrate the value of our proposed algorithm: (1) the short-term impact, (2) the long-term impact, and (3) the global treatment effect on advertising

revenue. This experiment, together with a comprehensive simulation study, shows that advertising revenue from our oSBL algorithm increased at least 5.35%. For a large-scale platform such as Platform O, such an increase would translate to hundreds of millions of US dollars per year.

6.1. Short-Term Performance of Our oSBL Algorithm

We present the model-free results here. See Appendix C.4 for the robustness checks with regression models. We base our analysis on the following metrics *during* the experiment period:

1. *Cold Start Success Rate.* This metric is defined as the proportion of ads whose total number of conversions exceeds the conversion target, i.e., $\frac{\sum_{j=1}^{K} \mathbb{I}_{\{V_j \geq \alpha T\}}}{K}$, where K is the total number of new ads assigned to the respective experimental group. For Platform O, the cold start period of any ad is the first three days, whereas the conversion target during the cold start period is ten conversions, i.e., $\alpha T = 10$. New ads that arrive in the last three days of the experiment do not pass the entire cold start period, but this will not affect comparisons between the treatment and control groups. For completeness, we also examine the cold start reward clustered at the ad level, i.e., $\beta_j \min\{V_j, \alpha T\}$ for each ad j.
2. *Short-Term Revenue.* This metric is clustered at the UV level for *all* (old and mature) ads in different experiment groups on the DSP.
3. *Ratio Between Real and Target Cost per Conversion.* This metric is clustered at the ad level to evaluate the impact of our oSBL algorithm on the controllability of advertisers' costs. If this ratio is significantly larger than 1, it implies that our new algorithm substantially increases the cost per conversion for the advertisers, which may cause them to complain or even leave the platform.

We summarize our experimental results on the short-term impact of oSBL algorithm in Table 2, which can also be replicated by regression analysis presented in Appendix C.4. It is evident from Panel A of the table that our oSBL algorithm has substantially increased both the cold start success rate and the cold start reward of new ads by 61.62% and 47.41% (p-values ≤ 0.0001). Such improvements result from the shadow bids produced by oSBL that are added to the real-time system bidding prices, which also give rise to a 44.94% increase in ad impressions, a 35.34% increase in ad clicks, and a 59.67% increase in ad conversions (p-values ≤ 0.01) during the cold start period of new ads.

Thanks to the shadow bids, our oSBL algorithm significantly improved the new-ad cold start performance, but it could also have cannibalized the impressions and conversions of mature ads. As a consequence, the algorithm could have reduced total short-term revenue during the experiment. Comparing the per-UV revenue of the treatment and control groups on the UV side, Panel B of Table 2 confirms this intuition, by quantifying that the oSBL algorithm will decrease short-term revenue by 0.717% (with a p-value less than 0.01). This small relative decrease in short-term

Table 2 The Short-Term Effects of oSBL

Panel A: Effects on the Cold Start at the Ad Level			
	Time Window: May 23–30, 2020		
	Treatment		Control
	(1)		(2)
Number of impressions	29,866 (268,139)		20,605 (232,143)
Relative effect size		44.94%****	
Number of clicks	2,352 (24,088)		1,738 (30,780)
Relative effect size		35.34%**	
Number of conversions	9.38 (108.93)		5.86 (68.67)
Relative effect size		59.67%****	
Observations	34,605		34,076
Cold start success rate	0.0438 (0.204)		0.0271 (0.162)
Relative effect size		61.62%****	
Cold start reward	261.6 (958.1)		177.1 (930.5)
Relative effect size		47.71%****	
Observations	34,605		34,076
Panel B: Effects of the Algorithm on Short-Term Revenue and the Objective Value			
	Time Window: May 23–30, 2020		
	Treatment		Control
	(1)		(2)
Revenue per user	1.439 (37.81)		1.448 (37.83)
Relative effect size of total revenue		-0.717%**	
Observations	240,308,309		240,538,298
Total objective value	354,856,325		354,334,315
Relative effect size		0.147%****	
Panel C: Effects of the Algorithm on Advertiser Costs			
$\frac{\text{Real Cost}}{\text{Target Cost}} - 1$	$-30\%\sim30\%$	$30\%\sim100\%$	$>100\%$
	(1)	(2)	(3)
Proportion of ads (treatment condition)	0.665	0.041	0.059
	(0.472)	(0.198)	(0.235)
Proportion of ads (control condition)	0.648	0.026	0.045
	(0.477)	(0.159)	(0.209)
p-value of t-test	0.74	0.45	0.58

Notes: Mean values are reported in this table. To protect sensitive data, the impressions, clicks, conversions, and revenues are linearly scaled. The cutoff ranges of Panel C—$[-30\%, 30\%]$, $[30\%, 100\%]$, and $> 100\%$—are adopted in consistency with Platform O's business practice. Standard errors in Panels A and C are clustered at the ad level and reported in parentheses. *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001.

revenue is both within our expectation and acceptable for Platform O. And, as we articulate in Section 6.3, a short-term loss (-0.717%) can be well compensated for by the long-term revenue boost (5.35%) of the oSBL algorithm.

Are the improvements in success rate and reward offset by increased advertiser costs? Panel C of Table 2 addresses this question by examining the distribution of the relative gap between the real cost of an ad and its target cost gap (measured by $\frac{\text{Real Cost}}{\text{Target Cost}} - 1$). It shows that there is no significant difference between the distribution of the relative gap for ads in the treatment group and that for ads in the control group. Specifically, the treatment and control groups are similar in the proportion of new ads whose relative cost gap is in each of the following ranges [-30%,30%], [30%,100%], and $> 100\%$. The results show that the oSBL algorithm does not boost advertisers' cost to increase the cold start success rate and the cold start reward of their new ads.

Last but not least, our oSBL algorithm has substantially increased the prediction accuracy for the CTR of new ads. Specifically, our two-sided experiments show that the AUC of new-ad CTR prediction in the treatment group is 7.48% larger than the one in the control condition, with the p-value of t-test being 0.017. (To protect sensitive data, we only report the relative difference here.)

Table 3 The Long-Term Effects of oSBL

Panel A: Effects of oSBL on Ads			
	Time Window: May 31–August 31, 2020		
	Treatment	Control	Relative Increase
	(1)	(2)	(3)
Retention days	10.20 (11.03)	9.89 (10.81)	3.13%**
log (Impressions)	8.02 (3.18)	7.46 (3.00)	****
99% Winsorized impressions	107,424 (379,251)	63,981 (242,250)	67.90%****
log (Revenue)	2.60 (2.69)	2.13 (2.56)	****
99% Winsorized revenue	485 (1,914)	362 (1,526)	34.02%****
CTR	0.054 (0.059)	0.049 (0.056)	11.14%****
CVR	0.023 (0.132)	0.024 (0.138)	$p > 0.1$
Bid prices	57.14 (62,62)	57.19 (61.30)	$p > 0.1$
Observations	34,605	34,076	

Panel B: Effects of oSBL on Advertiser Behaviors				
	Time Window: May 31–June 25, 2020			
	Dependent Variable:			
	Bidding Prices	Impressions	Conversions	Retention Days
	(1)	(2)	(3)	(4)
Treatment−Control	29.57	5568	6.35	-.0023
	(334.04)	(13180)	(20.10)	(0.073)
p-value	0.93	0.67	0.75	0.75
Industry fixed effects	Yes	Yes	Yes	Yes
Bidding price		Yes	Yes	Yes
Budget	Yes	Yes	Yes	Yes
Target strategy	Yes	Yes	Yes	Yes

Notes: Mean values are reported in Panel A. To protect sensitive data, all metrics are linearly scaled. For Panel B, we report only the coefficient and its standard error of the endogenous variable (i.e., the number of conversions during the cold start period and the experiment). Standard errors in Panel A (Panel B) are clustered at the ad (advertiser) level and reported in parentheses. *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001.

6.2. Long-Term Performance of Our oSBL Algorithm

We next examine the long-term impact of oSBL on both ads and advertisers *after* the cold start period, when the shadow bids are set to zero. Regarding ads, we evaluate how oSBL influences the lifetime performance of an ad after the cold start period by comparing the following postexperiment metrics of the treatment and control ads: (a) retention days (number of days that an ad is active after the cold start period), (b) lifetime number of impressions, (c) lifetime revenue, (d) CTR/CVR, and (e) average system bidding price. Because the distribution of impression and lifetime advertising revenue after cold start is heavy tailed, we perform the t-test after taking log-transformation of or winsorizing the revenue at the 99% level. Regarding advertisers, we investigate whether oSBL changes advertiser behaviors, especially their bid prices and the length of time they wish to keep their ads active on Platform O.

Panel A of Table 3 documents the effects on ads. The results show that our algorithm significantly increased—by 3.13%—the average number of active days and, thus, the average market thickness (defined as the average number of ads competing for each user impression). Figure 6 plots the scaled (to protect sensitive data) total number of ads in different experiment conditions that remained active each day after the experiment—it shows that our proposed algorithm significantly increased market thickness—by 7.21% on average—especially during the first two weeks after cold start.

Comparing the lifetime revenues of the treatment and control ads reveals further insights. We find that oSBL boosted postexperiment advertising revenue after cold start by 34.02%. This benefit

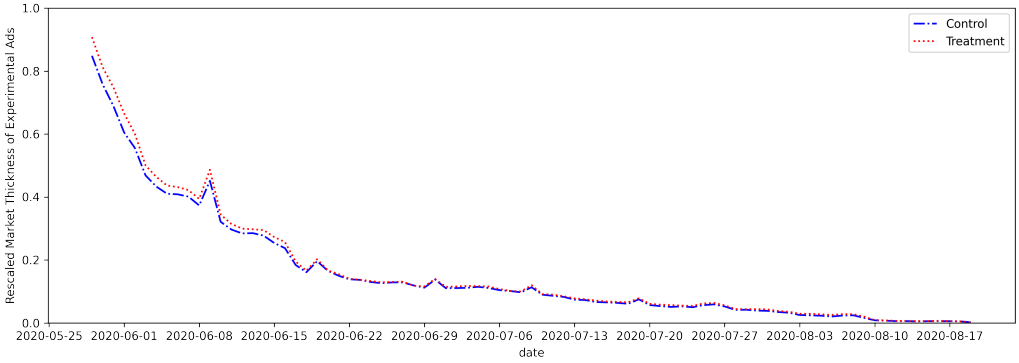


Figure 6 Effect of oSBL on Market Thickness

is driven by the fact that the algorithm not only thickens the market by retaining the ads longer but also successfully identifies high-quality ads with 11.14% higher CTRs. We also observe that our proposed algorithm had no significant impact on the CVR and average system bidding prices of an ad.

In summary, the oSBL algorithm substantially improved the market thickness and CTR of the ads after cold start. In Section 6.3, we build a simulation model to demonstrate that such long-term effects on ads could be translated into a significant global treatment effect of our algorithm on the advertising revenue of a DSP.

Given that the oSBL algorithm significantly improves ads' CTRs and, thus, revenue performance, would advertisers also respond to such improvements by changing their behaviors on Platform O (such as bidding prices)? In particular, if an advertiser increases its expectation on cold start performance, would the effectiveness of our algorithm be weakened? To adress these questions, we next examine whether advertisers would behave differently after oSBL is adopted. To this end, we adopt a two-stage least squares (2SLS) specification in Equation (10) to identify whether the total number of conversions during the cold start period will change an advertiser's behavior, where X_j are the advertiser-specific features such as industry fixed effects, bidding prices, budge, and target strategy. We define *exp ratio* as the proportion of treatment ads among all the ads in our experiment for each advertiser and adopt it as the instrumental variable. The endogenous variable is the total number of conversions by an advertiser during the cold start period and the experiment, whereas the dependent variable may take different forms such as average bidding prices, the total number of impressions/conversions, and the average number of retention days for all the ads of an advertiser. Note that exp ratio is a valid instrument in this setting. One the one hand, the p-values of the weak instrument tests are smaller than 10^{-5}, so the strong first-stage assumption holds.

On the other hand, it is unlikely that exp ratio could impact an advertiser's behavior through a channel other than conversions, so the exclusion restriction also holds.

$$\begin{aligned} &\text{First Stage: Cold Start Conversion}_j = \alpha_0 + \alpha_1 \text{exp ratio}_j + X_j + \epsilon \\ &\text{Second Stage: Dependent Variable}_j = \beta_0 + \beta_1 \text{Cold Start Conversion}_j + X_j + \epsilon \end{aligned} \tag{10}$$

Finally, Panel B of Table 3 reports the estimation results for the 4,340 advertisers we study. After controlling for industry fixed effects, bidding prices, budget, and target strategy, we find no evidence that implies our oSBL algorithm significantly changed advertisers' long-term behaviors on Platform O.

6.3. Global Treatment Effect of Our oSBL Algorithm on Advertising Revenue

Our experiment design cannot directly observe the effect of oSBL on the long-term revenue change, because all the experimental new ads would flow into the pool of mature ads and join the auction under both treatment and control UVs. One solution is to use the two-sided experiment with blocking for those experimental new ads after they mature. Though this design can better deal with cannibalization and spillover effects when estimating the long-term effect of the algorithm on the revenue, it is much more costly, due to the lifetime blocking of all ads for a substantial portion of UVs.

In this subsection, we seek to quantify, through simulation, the global treatment effect of our oSBL algorithm on advertising revenue. Specifically, based on the our empirical results on the long-term impact of the algorithm (Table 3, Panel A), our oSBL could substantially improve the length of ads' retention time by 3.13% and CTR by 11.14% without negatively affecting their CVR and average system bidding prices. This motivates us to build a simulation model to translate such positive impact into the long-term revenue boost for the platform.

To estimate the long-term revenue increase our algorithm could generate, we use data with 12 million impressions between April 9 and April 30, 2020. We randomly sampled 1.2 million impressions with replacement for each simulation and replicated 10 times via Bootstrap—the following results all pass the t-test with p-values smaller than 10^{-3}. In our simulation, we apply the oSBL for the new ads in the treatment condition. After the cold start, new ads flowed into the pool of mature ads. As we documented in Section 6.2, the ads under the oSBL have a higher CTR and longer retention after the cold start period. To model this oSBL effect, we assume the CTR of the treated ads will increase by Δ_{CTR} (relative changes) and their retention time length will increase by Δ_r (relative changes). As shown below, because the values of Δ_{CTR} and Δ_r may change once the algorithm is applied to all ads and the entire user traffic, we perform sensitivity analysis by varying the values of Δ_{CTR} and Δ_r.

We first validate our model by replicating our experimental results so that the short-term revenue will decrease when oSBL is applied to 20% of new ads (see Panel B of Table 2) during the experiment

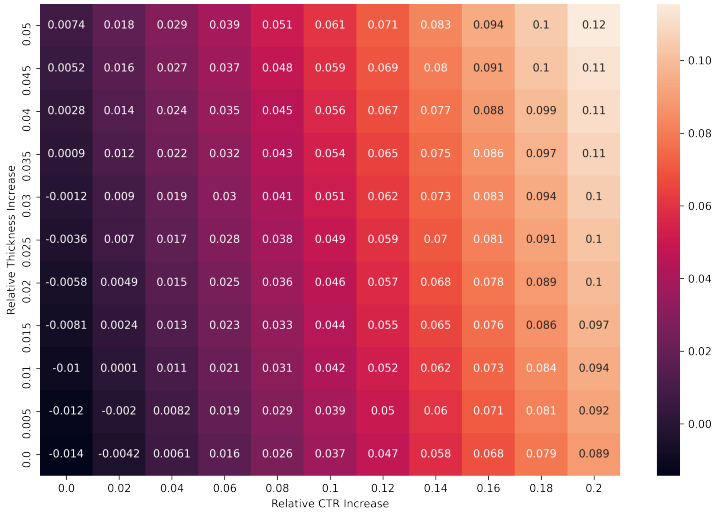


Figure 7 Global Treatment Effect of oSBL on Advertising Revenue

period (May 23–30, 2020). We use the nonexperiment impressions (B13, B23, B33, B14, B24, and B34 in Figure 5) in the simulation. We compare the total revenue of these eight days for two cases: where oSBL is applied to 20% of new ads and where the baseline algorithm is applied to all new ads. We find that the average short-term revenue decrease is 0.583%, which is consistent with our regression-based result that oSBL (applied to 20% of the ads) will decrease short-term revenue by 0.592%. Therefore, our simulation model is fairly accurate in predicting the short-term revenue loss caused by oSBL.

The estimation of two key parameters in our simulation model Δ_r (which refers to the average relative increase of the retention length for mature ads) and Δ_{CTR} (which refers to the average relative increase of the CTR for mature ads) relies on the two-sided experiment where only 20% of ads and 33% of UVs are included in the treatment group. Therefore, it is challenging to extrapolate their estimates to the counterfactual setting, where the oSBL algorithm is applied to the entire ad set and user population. To obtain a complete picture on the global treatment effect of our algorithm, we conduct a sensitivity analysis with our simulation model by varying Δ_r from 0 to 5% and varying Δ_{CTR} from 0 to 20%, assuming that oSBL is applied to all ads and UVs. Baseline revenue is denoted by R_0 and revenue under the oSBL algorithm is denoted by $R(\Delta_r, \Delta_{CTR})$ (so $R_0 = R(0,0)$). We are interested in the relative advertising revenue increase associated with oSBL:

$$\Xi(\Delta_r, \Delta_{CTR}) = \frac{R(\Delta_r, \Delta_{CTR}) - R_0}{R_0} \times 100\%$$

Figure 7 demonstrates that for a wide range of potential values for Δ_r and Δ_{CTR}, the relative

revenue increase $\Xi(\Delta_r, \Delta_{CTR})$ is significantly above 0, implying that our oSBL algorithm could substantially boost the long-term advertising revenue of a DSP. In particular, if we linearly extrapolate the estimates from our two-sided experiment, $\Delta_r = 3.13\%$ and $\Delta_{CTR} = 11.14\%$, to all ads and the entire population, the advertising revenue increase from our algorithm is at least 5.35%. For a large-scale platform such as Platform O, such an increase would translate to hundreds of millions of US dollars per year.

7. Discussion and Conclusion

We close by discussing several promising directions for future research. In Sections 3 and 5, we detail the cost-control problem brought by complicated auction mechanisms and bidding/payment methods in a real DSP. Future research could examine the cold start algorithm under real-time bidding. Furthermore, our cold start algorithm, in principle, could be embedded into a recommender system for general content as well. In this setting, ranking is a prominent data-driven decision. Thus, an interesting extension of our work would be integrating MAB algorithms and state-of-the-art ranking models such as Learning2Rank. Finally, future research could test other ways of conducting two-sided field experiments and quantify the biases that may be introduced by the violation of SUTVA in two-sided platforms.

References

Agarwal, Alekh, Miroslav Dudík, Satyen Kale, John Langford, Robert Schapire. 2012. Contextual bandit learning with predictable rewards. *Artificial Intelligence and Statistics*. PMLR, 19–26.

Agarwal, Alekh, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, Robert Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. *International Conference on Machine Learning*. 1638–1646.

Agrawal, Shipra, Nikhil R Devanur. 2014. Bandits with concave rewards and convex knapsacks. *Proceedings of the fifteenth ACM conference on Economics and computation*. ACM, 989–1006.

Agrawal, Shipra, Nikhil R Devanur, Lihong Li. 2016. An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. *Conference on Learning Theory*. 4–18.

Agrawal, Shipra, Zizhuo Wang, Yinyu Ye. 2014. A dynamic near-optimal algorithm for online linear programming. *Operations Research* **62**(4) 876–890.

Allen-Zhu, Zeyuan, Yuanzhi Li, Zhao Song. 2019. A convergence theory for deep learning via over-parameterization. *International Conference on Machine Learning*. PMLR, 242–252.

Arora, Sanjeev, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, Ruosong Wang. 2019. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*. 8139–8148.

Badanidiyuru, Ashwinkumar, Robert Kleinberg, Aleksandrs Slivkins. 2013. Bandits with knapsacks. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 207–216.

Balseiro, Santiago, Haihao Lu, Vahab Mirrokni. 2020. The best of many worlds: Dual mirror descent for online allocation problems.

Balseiro, Santiago R, Omar Besbes, Gabriel Y Weintraub. 2015. Repeated auctions with budgets in ad exchanges: Approximations and design. *Management Science* **61**(4) 864–884.

Balseiro, Santiago R, Jon Feldman, Vahab Mirrokni, Shan Muthukrishnan. 2014. Yield optimization of display advertising with ad exchange. *Management Science* **60**(12) 2886–2907.

Balseiro, Santiago R, Yonatan Gur. 2019. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science* **65**(9) 3952–3968.

Bastani, Hamsa, David Simchi-Levi, Ruihao Zhu. 2019. Meta dynamic pricing: Transfer learning across experiments. *Available at SSRN 3334629* .

Bharadwaj, Vijay, Peiji Chen, Wenjing Ma, Chandrashekhar Nagarajan, John Tomlin, Sergei Vassilvitskii, Erik Vee, Jian Yang. 2012. Shale: an efficient algorithm for allocation of guaranteed display advertising. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1195–1203.

Bietti, Alberto, Alekh Agarwal, John Langford. 2018. A contextual bandit bake-off. *arXiv preprint arXiv:1802.04064* .

Blake, Thomas, Dominic Coey. 2014. Why marketplace experimentation is harder than it seems: The role of test-control interference. *Proceedings of the fifteenth ACM conference on Economics and computation*. 567–582.

Boucheron, Stéphane, Gábor Lugosi, Pascal Massart. 2013. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press.

Caldentey, René, Gustavo Vulcano. 2007. Online auction and list price revenue management. *Management Science* **53**(5) 795–813.

Cao, Yuan, Quanquan Gu. 2019. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in Neural Information Processing Systems*. 10836–10846.

Chen, Boxiao, Xiuli Chao, Hyun-Soo Ahn. 2019. Coordinating pricing and inventory replenishment with nonparametric demand learning. *Operations Research* **67**(4) 1035–1052.

Chen, Ningyuan, Guillermo Gallego. 2018. A primal-dual learning algorithm for personalized dynamic pricing with an inventory constraint. *Available at SSRN 3301153* .

Chen, Weidong, Cong Shi, Izak Duenyas. 2020. Optimal learning algorithms for stochastic inventory systems with random capacities. *Production and Operations Management* .

Chizat, Lenaic, Edouard Oyallon, Francis Bach. 2019. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*. 2937–2947.

Choi, Hana, Carl F Mela, Santiago R Balseiro, Adam Leary. 2020. Online display advertising markets: A literature review and future directions. *Information Systems Research* .

Chu, Wei, Lihong Li, Lev Reyzin, Robert Schapire. 2011. Contextual bandits with linear payoff functions. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 208–214.

Covington, Paul, Jay Adams, Emre Sargin. 2016. Deep neural networks for youtube recommendations. *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

Cui, Ruomeng, Jun Li, Dennis Zhang. 2019a. Discrimination with incomplete information in the sharing economy: Evidence from field experiments on airbnb. *Management Science* Forthcoming.

Cui, Ruomeng, Dennis J Zhang, Achal Bassamboo. 2019b. Learning from inventory availability information: Evidence from field experiments on amazon. *Management Science* **65**(3) 1216–1235.

Dave, Kushal, Vasudeva Varma. 2014. Computational advertising: Techniques for targeting relevant ads. *Foundations and Trends in Information Retrieval* **8**(4–5) 263–418.

Devanur, Nikhil R, Thomas P Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. *Proceedings of the 10th ACM conference on Electronic commerce*. 71–78.

Dudik, Miroslav, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, Tong Zhang. 2011. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369* .

Feldman, Jake, Dennis J Zhang, Xiaofei Liu, Nannan Zhang. 2018. Taking asssortment optimization from theory to practice: Evidence from large field experiments on alibaba Working Paper.

Ferreira, Kris Johnson, David Simchi-Levi, He Wang. 2018. Online network revenue management using thompson sampling. *Operations research* **66**(6) 1586–1602.

Fisher, Marshall, Santiago Gallino, Jun Li. 2018. Competition-based dynamic pricing in online retailing: A methodology validated with field experiments. *Management science* **64**(6) 2496–2514.

Foster, Dylan, Alekh Agarwal, Miroslav Dudik, Haipeng Luo, Robert Schapire. 2018. Practical contextual bandits with regression oracles. *Proceedings of Machine Learning Research* **80**.

Gallego, Guillermo, Garrett Van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science* **40**(8) 999–1020.

Golrezaei, Negin, Adel Javanmard, Vahab Mirrokni. 2019. Dynamic incentive-aware learning: Robust pricing in contextual auctions. *Advances in Neural Information Processing Systems*. 9759–9769.

Golrezaei, Negin, Hamid Nazerzadeh, Paat Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science* **60**(6) 1532–1551.

Ha-Thuc, Viet, Avishek Dutta, Ren Mao, Matthew Wood, Yunli Liu. 2020. A counterfactual framework for seller-side a/b testing on marketplaces. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2296.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

Hojjat, Ali, John Turner, Suleyman Cetintas, Jian Yang. 2017. A unified framework for the scheduling of guaranteed targeted display advertising under reach and frequency requirements. *Operations Research* **65**(2) 289–313.

Hsu, Daniel, Sham M. Kakade, Tong Zhang. 2014. Random design analysis of ridge regression. *Foundations of Computational Mathematics* **14**(3) 569–600.

Imbens, Guido W, Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Jacot, Arthur, Franck Gabriel, Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*. 8571–8580.

Johari, Ramesh, Hannah Li, Gabriel Weintraub. 2020. Experimental design in two-sided platforms: An analysis of bias. *arXiv preprint arXiv:2002.05670* .

Li, Xiaocheng, Chunlin Sun, Yinyu Ye. 2020. Simple and fast algorithm for binary integer and online linear programming. *arXiv preprint arXiv:2003.02513* .

Nambiar, Mila, David Simchi-Levi, He Wang. 2019. Dynamic learning and pricing with model misspecification. *Management Science* **65**(11) 4980–5000.

Pouget-Abadie, Jean, Kevin Aydin, Warren Schudy, Kay Brodersen, Vahab Mirrokni. 2019. Variance reduction in bipartite experiments through correlation clustering. *Advances in Neural Information Processing Systems* **32** 13309–13319.

Rolnick, David, Kevin Aydin, Jean Pouget-Abadie, Shahab Kamali, Vahab Mirrokni, Amir Najmi. 2019. Randomized experimental design via geographic clustering. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2745–2753.

Schwartz, Eric M, Eric T Bradlow, Peter S Fader. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science* **36**(4) 500–522.

Simchi-Levi, David, Yunzong Xu. 2020. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Available at SSRN* .

Terwiesch, Christian, Marcelo Olivares, Bradley R Staats, Vishal Gaur. 2020. Om forum—a review of empirical operations management over the last two decades. *Manufacturing & Service Operations Management* **22**(4) 656–668.

Tropp, Joel A. 2015. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571* .

Valko, Michal, Nathaniel Korda, Rémi Munos, Ilias Flaounas, Nelo Cristianini. 2013. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869* .

Vartak, Manasi, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6907–6917.

Wager, Stefan, Guenther Walther. 2015. Adaptive concentration of regression trees, with application to random forests. *arXiv preprint arXiv:1503.06388* .

Yang, Lin, Mengdi Wang. 2020. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. *International Conference on Machine Learning*. PMLR, 10746–10756.

Yang, Xun, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, Kun Gai. 2019. Bid optimization by multivariable control in display advertising. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1966–1974.

Zeng, Zhiyu, Hengchen Dai, Dennis Zhang, Zuo-Jun Max Shen, Zhiwei Xu, Heng Zhang, Renyu Philip Zhang. 2020. Social nudges boost productivity on onlineplatforms: Evidence from field experiments. *Available at SSRN 3611571* .

Zhang, Dennis J, Hengchen Dai, Lingxiu Dong, Fangfang Qi, Nannan Zhang, Xiaofei Liu, Zhongyi Liu, Jiang Yang. 2020. The long-term and spillover effects of price promotions on retailing platforms: Evidence from a large randomized experiment on alibaba. *Management Science* **66**(6) 2589–2609.

Zhang, Heng, Paat Rusmevichientong, Huseyin Topaloglu. 2018. Multiproduct pricing under the generalized extreme value models with homogeneous price sensitivity parameters. *Operations Research* **66**(6) 1559–1570.

Zhang, Shuai, Lina Yao, Aixin Sun, Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* **52**(1) 1–38.

Zhang, Weinan, Yifei Rong, Jun Wang, Tianchi Zhu, Xiaofan Wang. 2016. Feedback control of real-time display advertising. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 407–416.

Zhou, Dongruo, Lihong Li, Quanquan Gu. 2020. Neural contextual bandits with ucb-based exploration. *International Conference on Machine Learning*.

Zhou, Guorui, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, Kun Gai. 2018. Deep interest network for click-through rate prediction. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

Online Appendices

Appendix A: Table of Notations

Table 4 Table of Notations

Notations in the Allocation Model and SBL Algorithm	
Notation	Description
K	The number of ads
$A=\{1,2,\dots,K\}$	The set of ads
T	The total number of the user views, namely, ad impressions
X	The finite or countably infinite context set
x_{tj}	x_{tj} is the feature vector associated with round t and ad j
$x_t=(x_{t1},\dots,x_{tK})\in X$	The context associated with round t
$\mathcal{I}=\{1,2,3,\dots\}$	The index set of context types
$a_t\in A$	The ad which is chosen to be displayed to the user view t
$v_t(a_t)\in\{0,1\}^K$	The K-dimensional binary vector representing whether each ad is clicked
$\mathcal{D}$	The distribution of $(x_t,\{v_t(a)\}_{a\in K})$ over $X\times\{0,1\}^{K\times K}$
$\mathcal{D}_X$	The marginal distribution of $\mathcal{D}$ over context types $[m]$
c_{ij}	The CTR of ad j under the context i
$V:=\sum_{t=1}^{T}v_t(a_t)$	The accumulated click-through vector
$b_j\in[0,1]$	The bid per click of ad $j\in A$
$\beta_j\in(0,1]$	The cold start reward per click of ad $j\in A$
$\alpha\in(0,1)$	The target click per round
$\Gamma(V)$	The objective value
$\mathcal{H}_t=\bigcup_{s=1,\dots,t-1}\{(x_s,a_s,v_s(a_s))\}$	The history update to round t
$\Delta_A=\{y\in\mathbb{R}^{\lvert A\rvert}:y_j\geq 0,\forall j\in A,\sum_{j\in A}y_j\leq 1\}$	The distribution over arms which defines the feasible ad allocation plan
π	The policy mapping from $\mathcal{H}_t$ to Δ_A
$\hat{c}_{ij}^t$	The predicted CTR based on $\mathcal{H}_t$ of ad j under context i
λ^{t*}	The empirically optimal shadow bidding prices at round t
λ^t	The shadow bidding prices generated by the SBL algorithm at round t
p_i	The probability that context $i\in\mathcal{I}$ occurs
$\hat{p}_i^t$	The empirically estimated probability that context $i\in\mathcal{I}$ occurs at round t
$\tau_1<\tau_2<\dots$	The epoch schedule to update λ
$f(x)=O(g(x))$	There exists a positive constant c such that $f(x)\leq c\cdot\lvert g(x)\rvert$ for sufficiently large x
$f(x)=\tilde{O}(g(x))$	There exists a positive constant c such that $f(x)\leq c\cdot\lvert g(x)\rvert\cdot\log^k(\lvert g(x)\rvert)$ for some $k>0$ and sufficiently large x
$f(x)=\Omega(g(x))$	There exists a positive constant c such that $f(x)\geq c\cdot\lvert g(x)\rvert$ for sufficiently large x
$f(x)=\Theta(g(x))$	$\lvert f(x)\rvert/\lvert g(x)\rvert$ converges to 1 as x goes to infinity.
Notations in the Neural Network Prediction Oracle	
Notation	Description
$\mathcal{X}$	The set of functions $(X\times A\mapsto[0,1])$ to estimate CTR
w_0	The dimension of the context $x_{ij}\in\mathbb{R}^{w_0}$
w	The number of hidden nodes of the neural network
L	The number of hidden layers of the neural network
d	The effective dimension of the regressor
$\theta\in\mathbb{R}^d$	The coefficients of parameters in the neural network
$\theta_0\in\mathbb{R}^d$	The initialized coefficients of parameters in the neural network
$\theta^t\in\mathbb{R}^d$	The updated coefficients of parameters in the neural network at round t
$H_j(x_{ij},\theta)$	The output of the neural network parameterized by θ given the feature input x_{ij} associated with context i and ad j
$\theta^*\in\mathbb{R}^d$	The coefficients such that $c_{ij}=\langle\nabla_\theta H_j(x_i,\theta_0),\theta^*-\theta_0\rangle$
λ_0	The regularization parameter in training the neural network
η	The step size in training the neural network
U	The number of descent steps in training the neural network
I_d	The identity matrix of dimension d
$\boldsymbol{H}$	The neural tangent kernel matrix defined by Zhou et al. (2020)
γ	The scalar such that $\boldsymbol{H}\succeq\gamma I$, where $M_1\succeq M_2$ refers to that M_1-M_2 is semi-positive-definite
Notations in Proofs	
Notation	Description
N_j^t	The set of contexts i for which $j\in\arg\max_{j'}\hat{c}_{ij'}^t(b_{j'}+\lambda_{j'})$
$\mathcal{T}_j^t$	The time periods before the round t when ad j is displayed
$n_j^t=\lvert\mathcal{T}_j^t\rvert$	The cardinality of the set $\mathcal{T}_j^t$, i.e, the number of displays of ad j before round t
$g_{ij}=\nabla_\theta H_j(x_i,\theta_0)$	The gradients of the function $H_j(x_i,\theta_0)$
$\hat{\gamma}_j^t=\sum_{i\in N_j^t}p_i\hat{c}_{ij}^t$	The empirical estimated probability of click-through for ad j at round t
$\gamma_j^t=\sum_{i\in N_j^t}p_i c_{ij}^t$	The expected probability of click-through for ad j at round t
$n(j)$	The number of times that ad j is clicked over the whole horizon

Appendix B: Supporting Argument for Regret Analysis

We devote this section to the proof of Theorem 1. Supporting analysis for justifying the prediction oracle assumption (Assumption 2) for the case of neural networks can be found in Appendix G. Before presenting the full-fledged proof of Theorem 1, we first give the proof of Lemma 1, followed by some preliminaries.

B.1. Proof of Lemma 1.

Let y^* denote the optimal solution of the optimization model in Lemma 1. Consider an arbitrary policy π, we have the following observation:

$$\begin{aligned}\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(V)\big] &= \frac{1}{T}\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{j=1}^K b_j\sum_{t=1}^T v_{tj}+\sum_{j=1}^K\beta_j\min\left\{\sum_{t=1}^T v_{tj},\alpha T\right\}\right]\\ &= \mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{j=1}^K b_j\sum_{t=1}^T v_{tj}/T+\sum_{j=1}^K\beta_j\min\left\{\sum_{t=1}^T v_{tj}/T,\alpha\right\}\right]\\ &\le \sum_{j=1}^K b_j\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right],\alpha\right\},\end{aligned}$$

in which the inequality follows from the Jensen's inequality.

Let us use $\mathcal{A}_{tj}$ to denote the event that ad j is displayed for user t and $\mathcal{D}_{ij}$ to denote the distribution that ad j is clicked when the context is i and ad j is displayed. It then follows that

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right]=\sum_{t=1}^T\mathbb{E}_{i\sim\mathcal{D}_X}[\mathbb{E}_{\mathcal{H}^t,\pi,\mathcal{D}_{ij}}[v_{tj}|i]/T]=\sum_{t=1}^T\mathbb{E}_{i\sim\mathcal{D}_X}[\mathbb{P}_\pi(\mathcal{A}_{tj}|i)/T\cdot c_{ij}]\\ =&\mathbb{E}_{i\sim\mathcal{D}_X}\left[\sum_{t=1}^T\mathbb{P}_\pi(\mathcal{A}_{tj}|i)/T\cdot c_{ij}\right]=\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi],\end{aligned}$$

in which we define $y_{ij}^\pi=\sum_{t=1}^T\mathbb{P}_\pi(\mathcal{A}_{tj}|i)/T$. Note that for any fixed i, it must be that $\sum_{j=1}^K y_{ij}^\pi=1$. Therefore,

$$\begin{aligned}\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(V)\big]&\le\sum_{j=1}^K b_j\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi],\alpha\right\},\\ &\le\sum_{j=1}^K b_j\mathbb{E}_{i\sim\mathcal{D}_X}\left[c_{ij}y_{ij}^*\right]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^*],\alpha\right\}=\mathsf{OPT}.\end{aligned}$$

This concludes the proof. □

B.2. Preliminaries for Regret Analysis

We first make several additional assumptions in our proof, purely for the ease and clarity of exposition. First of all, instead of solving

$$\mathsf{OPT}^t=\min_{\lambda_j\in[0,\beta_j],\ \forall j\in A}\sum_{i\in\mathcal{I}}\hat{p}_i^t\max_{j=1,2,\ldots,K}\left(\hat{c}_{ij}^t(b_j+\lambda_j)\right)+\alpha\sum_{j=1}^K(\beta_j-\lambda_j),\tag{11}$$

in the analysis, we assume that we solve

$$\mathsf{OPT}^t=\min_{\lambda_j\in[0,\beta_j],\ \forall j\in A}\sum_{i\in\mathcal{I}}p_i\max_{j=1,2,\ldots,K}\left(\hat{c}_{ij}^t(b_j+\lambda_j)\right)+\alpha\sum_{j=1}^K(\beta_j-\lambda_j).\tag{12}$$

That is, we assume that we observe p_i for any $i\in\mathcal{I}$, instead of only having access to its empirical estimate. In the meanwhile, we replace Assumption 1 with $p_i\hat{c}_{ij}\le O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{-\frac{5}{3}})$ for all $i\in\mathcal{I}$. This is without loss

of generality, because all the argument presented in this section still follows without this assumption. Indeed, with McDiarmid's inequality and the bound of Rademacher complexity term (Boucheron et al. 2013) with countably many contexts, we can still uniformly bound the error of the empirical probability estimate $\hat{p}_i^t$. Specifically, with probability at least $1-t^{-4}$, for any context $i \in \mathcal{I}$, we have $|\hat{p}_i^t - p_i| \le O(\sqrt{\log t/t})$, where t is the total number of occurrences for context i. As a result, this introduces a lower order error than $\tilde{O}(t^{-1/3})$, which can be ignored for our regret analysis. We will discuss more about this point at the end of the proof of Theorem 1 (see Appendix B.3).

Second, we assume that, when we solve (12), the inputs are in a *general position*. In other words, for any shadow bidding prices λ and round t when deciding which ad to display given a context, namely, $\left|\left\{i : |\arg\max_k\{\hat{c}_{ik}^t(\lambda_k + b_k)\}| > 1\right\}\right| \le K$. This assumption is introduced to avoid too many ties in Step 1 of the SBL algorithm, thus bounding the gap between primal and dual solutions in a lower order compared to the total regret. Similar assumptions are also made in the online matching and online linear program literature, e.g., Devanur and Hayes (2009), Agrawal et al. (2014). As argued by Devanur and Hayes (2009), when the general position assumption does not hold, an infinitesimal permutation ξ_{ij} can be added to each $\hat{c}_{ij}^t$ without affecting much of the objective function value for any λ, where ξ_{ij} is chosen independently and uniformly at random from a tiny interval $[-\varepsilon, \varepsilon]$ with ε being arbitrarily small. Hence, it is without loss of generality to assume the total number of ties is bounded by K with probability one. As will be clear in the proof of Lemma 2 below, the general position assumption helps us bound the total number of entries for the allocation decision constructed from the dual that are different from the primal solution by $O(K^2)$.

Next, some definitions and notations are in order. Throughout the proof of Theorem 1, we define the reward process (for any policy) $\{r(x_t, a_t)\}_{t=1}^T$, where the reward at round t is denoted by:

$$r(x_t, a_t) = \begin{cases} 0, & \text{if } \ell_t(a_t) = 0, \\ b_{a_t}, & \text{if } \sum_{s=1}^{t-1} \ell_s(a_t) \ge \alpha T \text{ and } \ell_t(a_t) = 1, \\ b_{a_t} + \beta_{a_t}, & \text{if } \sum_{s=1}^{t-1} \ell_s(a_t) < \alpha T \text{ and } \ell_t(a_t) = 1, \end{cases}$$

where $\ell_t(a) = 1$ if and only if $a = a_t$ and a click-through occurs. Notice that the objective value is given by $\Gamma(V) = \sum_{t=1}^T r(x_t, a_t)$. By Lemma 1, the expected reward satisfies $\mathbb{E}[\sum_{t=1}^T r(x_t, a_t)] \le T \cdot \mathsf{OPT}$. We note that $r(\cdot, \cdot)$ corresponds to the real reward collection process, which is hard to work with, because it depends on the click-through history of each ad so far. To overcome this challenge, we instead work with an auxiliary reward process, defined by

$$\tau(x_t, a_t) = \begin{cases} 0, & \text{if } \ell_t(a_t) = 0, \\ b_{a_t} + \beta_{a_t}, & \text{if } \ell_t(a_t) = 1. \end{cases}$$

Note $\tau(x_t, a_t) - r(x_t, a_t) = \beta_{a_t}$ when $\sum_{s=1}^{t-1} \ell_s(a_t) \ge \alpha T$ and a click-through occurs in round t. Otherwise, $\tau(x_t, a_t) = r(x_t, a_t)$.

Furthermore, we define N_j^t as the set of contexts i for which ad j is chosen to be displayed, with tie-breaking resolved, in round t. Thus, if $i \in N_j^t$, it holds that $j \in \arg\max_{j'} \hat{c}_{ij'}^t(b_{j'} + \lambda_{j'})$. Let λ^{t*} be the empirical optimal dual solution to (12), y^{t*} be the corresponding empirical optimal primal solution, and we use $\hat{y}$ to represent the primal integer allocation decision to any shadow bidding strategy λ with arbitrary tie-breaking. Hence, $N_j^t = \{i : \hat{y}_{ij}^t = 1\}$.

Before formally presenting the full-fledged analysis, we need to bound the regret induced by the tie-breaking problem and dual mirror descent in SBL algorithms. Clearly, complementary slackness implies that if $y_{i\ell}^{t*} > 0$,

then $\ell \in \arg\max_j \hat{c}^t_{ij}(b_j + \lambda^t_j)$. Hence, if $\arg\max_j \hat{c}^t_{ij}(b_j + \lambda^t_j)$ returns a unique solution ℓ, then the optimal primal allocation y^{t*} and the integer solution $\hat{y}^{t*}$ constructed from the dual λ^{t*} are the same. Because we use the dual-based solution to make the ad allocation decision in the primal space, the tie-breaking in the SBL algorithm can induce a difference between empirically optimal objective value and the objective value by the dual-based allocation. One may expect that the solution to (12) still yields a good performance due to complementary slackness, the general position assumption, and Assumption 1. Formally, the following lemma holds.

LEMMA 2 **(Approximate Complementary Slackness)**. *Under Assumption 1, we have:*

(a) Suppose the SBL-RS algorithm is applied. There exist a family of non-negative constants $\{\eta_j \geq 0 : j \in A\}$ *with* $\sum_{j\in A}\eta_j \leq O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})$, *such that the following approximate complementary slackness condition holds for each* $j \in A$*: (i) If* $\lambda^t_j \in [0, \beta_j)$, *we have* $\sum_{i\in N^t_j} p_i\hat{c}^t_{ij} \geq \alpha - \eta_j$. *(ii) If* $\lambda^t_j \in (0, \beta_j]$, *we have* $\sum_{i\in N^t_j} p_i\hat{c}^t_{ij} \leq \alpha + \eta_j$.

(b) Suppose the SBL-DMD algorithm is applied. Define $s_t(\lambda) = -\sum_{j\in[K]\setminus a_t}\alpha\lambda_j + (\hat{c}^t_{i_t a_t} - \alpha)\lambda_{a_t}$. *There exist a family of non-negative constants* $\{\eta_j \geq 0 : j \in A\}$ *with* $\sum_{j\in A}\eta_j \leq O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}) + \mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda) + \frac{2\eta}{\sigma} + \frac{1}{\eta}D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta}D_\varphi(\lambda, \lambda^{t+1})\right]$ *such that for all feasible* λ, *the approximate complementary slackness condition defined in part (a) holds.*

Proof of Lemma 2.

We first prove **Part (a)**, i.e., the approximate complementary slackness for the SBL-RS algorithm. To analyze the nonsmooth convex dual (9), we first write down the corresponding primal linear program (13) and its dual (14) as follows,

$$\begin{aligned} \text{(Primal)}\quad \mathsf{OPT}^t = \max_{y\geq 0, u\geq 0}\quad & \sum_{i\in\mathcal{I}}\sum_{j=1}^{K} p_i\hat{c}^t_{ij}b_j y_{ij} + \sum_{j=1}^{K}\beta_j(\alpha - u_j) \\ \text{s.t.}\quad & \sum_{j=1}^{K} y_{ij} \leq 1,\ \forall i \in \mathcal{I},\ \sum_{i\in\mathcal{I}} p_i\hat{c}^t_{ij}y_{ij} + u_j \geq \alpha,\ \forall j \leq K, \end{aligned} \tag{13}$$

and,

$$\begin{aligned} \text{(Dual)}\quad \mathsf{OPT}^t = \min_{\lambda\geq 0, \mu}\quad & \sum_{i\in\mathcal{I}} p_i\mu_i + \alpha\sum_{j=1}^{K}(\beta_j - \lambda_j) \\ \text{s.t.}\quad & \lambda_j \leq \beta_j\ \forall j \leq K \\ & \mu_i - \hat{c}^t_{ij}\lambda_j \geq \hat{c}^t_{ij}b_j\ \forall i \in \mathcal{I},\ \forall j \leq K. \end{aligned} \tag{14}$$

Let (y^{t*}, u^{t*}) and (λ^{t*}, μ^{t*}) be the optimal solution to (13) and (14) respectively. Clearly, the following complementary slackness conditions

$$\lambda^{t*}_j(\alpha - u^{t*}_j - \sum_{i\in\mathcal{I}} p_i\hat{c}^t_{ij}y^{t*}_{ij}) = 0,\ u^{t*}_j(\lambda^{t*}_j - \beta_j) = 0,\ \text{and}\ y^{t*}_{ij}(\mu^{t*}_i - \hat{c}^{t*}_{ij}\lambda^{t*}_j - \hat{c}^{t*}_{ij}b_j) = 0,$$

hold for all $i \in \mathcal{I}$ and $j \in A$. To highlight the intuition, let us first consider the case in which there is no tie in $\arg\max_{j'}\hat{c}^t_{ij'}(b_{j'} + \lambda^{t*}_{j'})$ for any $i \in \mathcal{I}$. Notice that if $\ell \notin \arg\max_{j'}\hat{c}^t_{ij'}(b_{j'} + \lambda^{t*}_{j'})$, then the complementary condition implies that $y^{t*}_{i\ell} = 0$. Since the primal program is increasing in y, it must be that $y^{t*}_{ij} = 1$ if

$j = \arg\max_{j'} \hat{c}^t_{ij'}(b_{j'} + \lambda^{t*}_{j'})$. In this case, $i \in N^t_j$ if and only if $y^{t*}_{ij} = 1$, and $y^{t*}_{ij} = 0$ otherwise . If $\lambda^{t*}_j < \beta_j$, then $u^{t*}_j = 0$. As a result, $\sum_{i\in N^t_j} p_i \hat{c}^t_{ij} = \sum_{i\in N^t_j} p_i \hat{c}^t_{ij} y^{t*}_{ij} = \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^{t*}_{ij} + u^{t*}_j \geq \alpha$. Similarly, if $\lambda^{t*}_j > 0$, then $\alpha - u^{t*}_j - \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^{t*}_{ij} = 0$, which implies that $\sum_{i\in N^t_j} p_i \hat{c}^t_{ij} \leq \alpha$. So, if there is no tie, the lemma holds true with $\eta_j = 0$, $\forall j \in A$.

Next, we bound the gap in the case under tie-breaking under the primal allocation induced by the empirical optimal dual solution, i.e., gap induced by difference between $\hat{y}^{t*}$ and y^{t*}. By the general position assumption, there are at most K ties and, thus, the tie-breaking introduces at most K^2 different entries between a primal optimal allocation y^{t*} and the corresponding integer solution $\hat{y}^{t*}$. This is because, with an argument similar to the one in the previous argument, for $i \in \mathcal{I}$ such that there is no tie, the K-dimension decision vector must satisfy $\hat{y}^{t*}_{i\cdot} = y^{t*}_{i\cdot}$ and all entries in the vector belong to the set $\{0, 1\}$. Hence,

$$\begin{aligned}
&\sum_{j=1}^{K} \left| \sum_{i\in N^{t*}_j} p_i \hat{c}^t_{ij} - \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^{t*}_{ij} \right| = \sum_{j=1}^{K} \left| \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} \hat{y}^{t*}_{ij} - \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^{t*}_{ij} \right| \\
\leq &\sum_{j=1}^{K} \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} \left| \hat{y}^{t*}_{ij} - y^{t*}_{ij} \right| \leq O(K^2 (T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{-\frac{5}{3}})) = O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}),
\end{aligned}$$

in which the first inequality follows from the definition of $\hat{y}^{t*}$ and the second inequality is due to Assumption 1. Let us define $\eta_j := |\sum_{i\in N^{t*}_j} p_i \hat{c}^t_{ij} - \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^{t*}_{ij}|$. If $\lambda^{t*}_j < \beta_j$ (i.e., Part (i) of the approximate complementary slackness condition), it holds that $u^{t*}_j = 0$ and $\sum_{i\in N^{t*}_j} p_i \hat{c}^t_{ij} y^{t*}_{ij} = \sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^{t*}_{ij} + u^{t*}_j \geq \alpha$. Therefore, $\sum_{i\in N^{t*}_j} p_i \hat{c}^t_{ij} \geq p_i \hat{c}^t_{ij} y^{t*}_{ij} - \eta_j \geq \alpha - \eta_j$. The argument for the case $\lambda^{t*}_j > 0$ (i.e., Part (ii) of the approximate complementary slackness condition) follows similarly. This complete the proof of Part (a).

Finally, we prove **part (b)**, i.e., to bound the regret induced by dual mirror descent in SBL-DMD by using a standard result on online mirror descent, i.e., Proposition 1 in Section B.4. We define the primal objective function $\mathsf{Obj}^t(y, u)$ of the optimization model (13), and the dual objective function $\mathsf{Obj}^t(\lambda) := \sum_{i\in\mathcal{I}} p_i \max_{j'=1,2,\ldots,K} \left(\hat{c}^t_{ij'}(b_{j'} + \lambda_{j'}) \right) + \alpha \sum_{j=1}^{K} (\beta_j - \lambda_j)$ after using the optimal $\mu^t_i = \max_{j'\in[K]} \hat{c}^t_{ij'}(b_{j'} + \lambda_{j'})$, for all $i \in \mathcal{I}$, and $s_t(\lambda) = -\sum_{j\in[K]\setminus a_t} \alpha\lambda_j + (\hat{c}^t_{i_t a_t} - \alpha)\lambda_{a_t}$. In SBL-DMD, we update λ^t over periods and (y^t, u^t) denotes the corresponding primal decision, i.e., $(y^t, u^t) = \arg\max_{y\geq 0, u\geq 0} \sum_{i\in\mathcal{I}} \sum_{j=1}^{K} p_i \hat{c}^t_{ij} b_j y_{ij} + \sum_{j=1}^{K} \beta_j(\alpha - u_j) + \sum_{i\in\mathcal{I}} \mu^t_i (1 - \sum_{j=1}^{K} y_{ij}) + \sum_{j=1}^{K} \lambda^t_j (\sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y_{ij} + u_j - \alpha)$. One can check, for any realized i_t at perid t, the corresponding played arm a_t with primal decision $y^t_{i_t a_t} > 0$ is exactly the decision $a_t = \arg\max_{j\in[K]} \hat{c}^t_{i_t j}(b_j + \lambda^t_j)$ defined in SBL-DMD. Then, we have the following inequality,

$$\begin{aligned}
\mathsf{Obj}^t(y^{t*}, u^{t*}) - \mathsf{Obj}^t(y^t, u^t) &\leq O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}) + \mathsf{Obj}^t(y^{t*}, u^{t*}) - \left(\mathsf{Obj}^t(\lambda^t) - \sum_{j=1}^{K} \lambda^t_j (\sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^t_{ij} - \alpha) \right) \\
&\leq O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}) + \sum_{j=1}^{K} \lambda^t_j (\sum_{i\in\mathcal{I}} p_i \hat{c}^t_{ij} y^t_{ij} - \alpha) \\
&= O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}) + \mathbb{E}_{i\sim\mathcal{D}_X} \left[s_t(\lambda^t) \right] \\
&\leq O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}) + \mathbb{E}_{i\sim\mathcal{D}_X} \left[s_t(\lambda) + \frac{2\eta}{\sigma} + \frac{1}{\eta} D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta} D_\varphi(\lambda, \lambda^{t+1}) \right],
\end{aligned}$$

where the first inequality follows from tie-breaking error induced by the primal (y^t, u^t) and dual λ^t shown in Part (a), as well as the definition of the Lagrange dual. The second inequality follows from the weak duality,

the equality follows from the definition of $s_t(\lambda^t)$, and the third inequality from Proposition 1, which holds for any λ. Thus, we complete the proof. □

B.3. Proof of Theorem 1

In this part, we present the main argument for the proof of Theorem 1. Note that N_j^t and $\hat{c}_{ij}^t$ are random variables measurable with respect to the history $\mathcal{H}_t$. We define $\hat{\gamma}_j^t := \sum_{i\in N_j^t} p_i \hat{c}_{ij}^t$ and $\gamma_j^t := \sum_{i\in N_j^t} p_i c_{ij}^t$. One can show that for the SBL algorithms, the expected number of any ad j being sampled before round t is $\sum_{s=1}^t \frac{\epsilon_s}{K} = \Theta\left(t^{\frac{2}{3}}K^{-\frac{2}{3}}(\log t)^{\frac{1}{3}}\right)$. By Hoeffding's inequality, by round t, ad j has been sampled $\Theta\left(t^{\frac{2}{3}}K^{-\frac{2}{3}}(\log t)^{\frac{1}{3}}\right)$ times with probability $1-t^{-4}$. This implies that, by Assumption 2 and the union bound, the CTR estimate satisfies $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ads with probability at least $1-t^{-3}$. Combining the above observations, we will show the following lemma.

LEMMA 3 **(Per Period Gap of the Alternative Reward Process)**.

(*a*) *Conditioned on exploitation at round t of the SBL-RS algorithm, it holds that*

$$\mathbb{E}\left[\tau(x_t,a_t)\right] \geq \mathsf{OPT} + \mathbb{E}[\sum_{j=1}^K \beta_j(\gamma_j^t-\alpha)^+] - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right). \tag{15}$$

(*b*) *Conditioned on exploitation at round t of the SBL-DMD algorithm, it holds that, for all feasible λ*

$$\begin{aligned}\mathbb{E}\left[\tau(x_t,a_t)\right] \geq \mathsf{OPT} + \mathbb{E}[\sum_{j=1}^K \beta_j(\gamma_j^t-\alpha)^+] - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)\\ - \mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda) + \frac{2\eta}{\sigma} + \frac{1}{\eta}D_\varphi(\lambda,\lambda^t) - \frac{1}{\eta}D_\varphi(\lambda,\lambda^{t+1})\right].\end{aligned} \tag{16}$$

Proof of Lemma 3

We first show **part (b)**. Applying the approximate complementary slackness (Lemma 2(b)), we bound the expected empirical auxiliary reward process under the implementation of the dual-solution in the primal space:

$$\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t (b_j+\beta_j).$$

Fixing the history $\mathcal{H}_t$ and an arbitrary ad j, we have the following equality:

$$\begin{aligned}\mathsf{OPT}^t &= \sum_{i\in\mathcal{I}} p_i \max_{j'=1,2,\ldots,K}\left(\hat{c}_{ij'}^t(b_{j'}+\lambda_{j'}^t)\right) + \alpha\sum_{j=1}^K(\beta_j-\lambda_j^t)\\ &= \sum_{j=1}^K\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t(b_j+\lambda_j^t) + \alpha\sum_{j=1}^K(\beta_j-\lambda_j^t)\\ &= \sum_{j=1}^K\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t(b_j+\beta_j) - \sum_{j=1}^K(\hat{\gamma}_j^t-\alpha)(\beta_j-\lambda_j^t),\end{aligned}$$

where the first equality follows from the definition of OPT^t, the second from the definition of N_j^t, and the third from the identity $\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t = \hat{\gamma}_j^t$. Thus, we have

$$\sum_{j=1}^K\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t(b_j+\beta_j) = \mathsf{OPT}^t + \sum_{j=1}^K(\hat{\gamma}_j^t-\alpha)(\beta_j-\lambda_j^t). \tag{17}$$

Hence, if the (exact) complementary slackness condition holds with $\eta_j = 0$ for all $j \in A$ (see Lemma 2(b)), we have $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = \beta_j(\hat{\gamma}_j^t - \alpha)^+$. In this case,

$$\sum_{j=1}^{K} \sum_{i \in N_j^t} p_i \hat{c}_{ij}^t (b_j + \beta_j) = \mathsf{OPT}^t + \sum_{j=1}^{K} (\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = \mathsf{OPT}^t + \sum_{j=1}^{K} \beta_j (\hat{\gamma}_j^t - \alpha)^+.$$

Otherwise, $\eta_j > 0$ for some $j \in A$ in Lemma 2(b). In this case, we show the following bound for the SBL-DMD algorithm:

$$\sum_{j=1}^{K} (\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \sum_{j=1}^{K} \beta_j (\hat{\gamma}_j^t - \alpha)^+ - O\left(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}\right) - \mathbb{E}_{i \sim \mathcal{D}_X}\left[s_t(\lambda) + \frac{2\eta}{\sigma} + \frac{1}{\eta} D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta} D_\varphi(\lambda, \lambda^{t+1}) \right]. \tag{18}$$

To obtain the inequality in (18), we observe, by Lemma 2(b), that $\sum_{j \in A} \eta_j \leq O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}) + \mathbb{E}_{i \sim \mathcal{D}_X}\left[s_t(\lambda) + \frac{2\eta}{\sigma} + \frac{1}{\eta} D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta} D_\varphi(\lambda, \lambda^{t+1}) \right]$, so it suffices to show that

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \beta_j (\hat{\gamma}_j^t - \alpha)^+ - \eta_j \text{ for all } j \in A.$$

More specifically, we consider three cases: (a) $\lambda_j^t = 0$, (b) $\lambda_j^t \in (0, \beta_j)$, and (c) $\lambda_j^t = \beta_j$.

If $\lambda_j^t = 0$, we have $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = \beta_j(\hat{\gamma}_j^t - \alpha)$. If $\hat{\gamma}_j^t > \alpha$, clearly, $\beta_j(\hat{\gamma}_j^t - \alpha)^+ = \beta_j(\hat{\gamma}_j^t - \alpha)$. Otherwise, $(\hat{\gamma}_j^t - \alpha)^+ = 0$, and $\lambda_j^t = 0$ implies that $\eta_j \geq \alpha_j - \hat{\gamma}_j^t$. Therefore,

$$\beta_j(\hat{\gamma}_j^t - \alpha) = \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j(\alpha - \hat{\gamma}_j^t) \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j \eta_j \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j,$$

where the last inequality follows from $\beta_j \in [0, 1]$.

If $\lambda_j^t = \beta_j$, we have $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = 0$. Furthermore, the following inequality holds

$$0 = \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j(\hat{\gamma}_j^t - \alpha)^+ \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j \eta_j \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j,$$

where the first inequality follows from $\hat{\gamma}_j^t - \alpha \leq \eta_j$ (see Lemma 2(b)) and $\eta_j \geq 0$, which together imply $(\hat{\gamma}_j^t - \alpha)^+ \leq \eta_j$, and the second from $\beta_j \in [0, 1]$. It then follows that $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j$ for the case where $\lambda_j^t = \beta_j$.

If $\lambda_j \in (0, \beta_j)$, Lemma 2(b) suggests that $|\hat{\gamma}_j^t - \alpha| \leq \eta_j$. In the case where $\hat{\gamma}_j^t > \alpha$, we have $\hat{\gamma}_j^t - \alpha \leq \eta_j$ and, therefore,

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = (\hat{\gamma}_j^t - \alpha)^+ \beta_j - (\hat{\gamma}_j^t - \alpha)\lambda_j^t \geq (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j \lambda_j^t \geq (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j,$$

where the first inequality follows from $\hat{\gamma}_j^t - \alpha \leq \eta_j$, and the second from $\lambda_j^t < \beta_j \leq 1$. In the case where $\hat{\gamma}_j^t \leq \alpha$, we have

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = (\hat{\gamma}_j^t - \alpha)^+ \beta_j - (\alpha - \hat{\gamma}_j^t)(\beta_j - \lambda_j^t) \geq (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j \beta_j \geq (\hat{\gamma}_j^t - \alpha)^+ \beta_j - \eta_j,$$

where the first equality follows from $(\hat{\gamma}_j^t - \alpha)^+ = 0$, the first inequality from $0 \leq \beta_j - \lambda_j^t \leq \beta_j$ and $0 \leq \alpha - \hat{\gamma}_j^t \leq \eta_j$, and the second inequality from $\beta_j \in [0, 1]$. Therefore, $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j$ for all $j \in A$ and, hence, inequality (18) follows.

Finally, we evaluate $\mathbb{E}[\tau(x_t, a_t) | \mathcal{H}_t]$ and bound the terms OPT^t and $(\hat{\gamma}_j^t - \alpha)^+$. Consider two cases: (a) $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}} (\log t)^{\frac{1}{3}} K^{\frac{1}{3}} d^{\frac{1}{2}}\right)$ for all j, which occurs with probability at least $1 - t^{-3}$ (see the discussions

before Lemma 3); and (b) $|\hat{c}_{ij}^t - c_{ij}^t| \neq O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for some $j \in A$, which occurs with probability less than t^{-3}.

We first consider the case where $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ad j (which occurs with probability at least $1-t^{-3}$). It follows from the definition of OPT (see Lemma 1) that

$$\mathsf{OPT} = \min_{0\leq\lambda_j\leq\beta_j,\forall j\in A}\sum_{i\in\mathcal{I}}p_i\max_{j=1,2,\ldots,K}\left(c_{ij}(b_j+\lambda_j)\right)+\alpha\sum_{j=1}^{K}(\beta_j-\lambda_j).$$

Because $|\hat{c}_{ij}^t - c_{ij}| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$, by the definitions of OPT and OPT^t, we have

$$|\mathsf{OPT}^t - \mathsf{OPT}| \leq O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{19}$$

Similarly, we bound $\mathbb{E}[\tau(x_t,a_t)|\mathcal{H}_t]$ by

$$\mathbb{E}[\tau(x_t,a_t)|\mathcal{H}_t] = \sum_{j=1}^{K}\sum_{i\in N_j^t}p_i c_{ij}(b_j+\beta_j) \geq \sum_{j=1}^{K}\sum_{i\in N_j^t}p_i\hat{c}_{ij}^t(b_j+\beta_j) - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{20}$$

Furthermore, by Jensen's inequality, we observe that

$$(\hat{\gamma}_j^t-\alpha)^+ \geq \left(\gamma_j^t-\alpha-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)\sum_{i\in N(j)}p_i\right)^+ \geq (\gamma_j^t-\alpha)^+ - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{21}$$

Collecting the terms of (17), (18), (19), (20), and (21) above, we have that

$$\begin{aligned}\mathbb{E}\left[\tau(x_t,a_t)\middle|\mathcal{H}_t\right] \geq \mathsf{OPT} &+ \sum_{j=1}^{K}\beta_j(\gamma_j^t-\alpha)^+ - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)\\ &- \mathbb{E}_{i\sim\mathcal{D}_X}\left[s_t(\lambda)+\frac{2\eta}{\sigma}+\frac{1}{\eta}D_\varphi(\lambda,\lambda^t)-\frac{1}{\eta}D_\varphi(\lambda,\lambda^{t+1})\right].\end{aligned} \tag{22}$$

For the case $|\hat{c}_{ij}^t - c_{ij}^t| \neq O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$, which occurs with probability less than t^{-3}, we can bound the expected gap between $\mathbb{E}[\tau(x_t,a_t)]$ and OPT by $O(t^{-3})$, which is a lower order term compared to the gap in the case where $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. Integrating over the distribution of history $\mathcal{H}_t$, we have established inequality (16) for the SBL-DMD algorithm.

To **part (a)**, we adopt the same argument as the proof of inequality (18) together with Lemma 2(a) to show that, under the SBL-RS algorithm, the following inequality holds:

$$\sum_{j=1}^{K}(\hat{\gamma}_j^t-\alpha)(\beta_j-\lambda_j^t) \geq \sum_{j=1}^{K}\beta_j(\hat{\gamma}_j^t-\alpha)^+ - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right). \tag{23}$$

Hence, for the case $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ad j, the inequalities (17), (23), (19), (20), and (21) together imply that

$$\mathbb{E}\left[\tau(x_t,a_t)\middle|\mathcal{H}_t\right] \geq \mathsf{OPT} + \sum_{j=1}^{K}\beta_j(\gamma_j^t-\alpha)^+ - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right).$$

Therefore, similar to the proof of part (a), integrating over $\mathcal{H}_t$ will establish inequality (15) for the SBL-RS algorithm. □

Summing inequality (15) over the whole T periods (i.e., $t=1,2,...,T$) and invoking Jensen's Inequality, we have the following regret bound on the expected auxiliary reward process $\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]$ under the SBL-RS algorithm:

$$\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]\geq T\cdot\mathsf{OPT}+\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right]-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{24}$$

Similarly, summing inequality (16) over the whole T periods and invoking Jensen's Inequality implies that under the SBL-DMD algorithm:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]\geq T\cdot\mathsf{OPT}+\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right]-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)\\ -\mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right]-\frac{2\eta}{\sigma}T-\frac{1}{\eta}D_{\varphi}(\lambda,\lambda^1).\end{aligned} \tag{25}$$

Next, we bound the difference between the auxiliary reward process $\tau(x_t,a_t)$ and the true reward process $r(x_t,a_t)$. Denote the total number of clicks for ad j in all T rounds as $n(j)$.

LEMMA 4 **(Difference between Two Reward Processes)**. *Under both SBL-RS and SBL-DMD algorithms, it holds that*

$$\mathbb{E}\left[\sum_{j=1}^{K}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+\right]\leq O\left(\sqrt{KT\log T}\right). \tag{26}$$

Proof of Lemma 4.

It suffices to establish a high probability bound: With probability at least $1-T^{-4}$, the following inequality holds:

$$\sum_{j=1}^{K}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+\leq O\left(\sqrt{KT\log T}\right). \tag{27}$$

We now show that for any subset of ads, denoted by $\mathcal{S}$,

$$\sum_{j\in\mathcal{S}}n(j)-\sum_{j\in\mathcal{S}}\sum_{t=1}^{T}\gamma_j^t\leq O\left(\sqrt{KT\log T}\right)\text{ with probability at least }1-T^{-4}. \tag{28}$$

Note that given the history by round t, $\mathcal{H}_t$, the expected total number of clicks for ad j is given by γ_j^t. One can use Azuma-Hoeffding inequality to show that for a fixed subset $\mathcal{S}$, we have with probability at most T^{-4K},

$$\sum_{j\in\mathcal{S}}n(j)-\sum_{j\in\mathcal{S}}\sum_{t=1}^{T}\gamma_j^t\geq O\left(\sqrt{KT\log T}\right).$$

Take a union bound over all subsets and notice that $2^KT^{-4K}\leq T^{-4}$. Hence, (28) holds with probability at least $1-T^{-4}$.

We now show (27) by contradiction. Suppose that (27) does not hold with probability at least T^{-4}. Define $\mathcal{S}'$ as the set of ads such that $n(j)>\sum_{t=1}^{T}\gamma_j^t$. It follows that

$$\sum_{j\in\mathcal{S}'}n(j)-\sum_{j\in\mathcal{S}'}\sum_{t=1}^{T}\gamma_j^t=\sum_{j\in\mathcal{S}'}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)=\sum_{j=1}^{K}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+>O\left(\sqrt{KT\log T}\right),$$

with probability at least T^{-4}, which contradicts inequality (28). Thus, inequality (27) holds with probability at least $1-T^{-4}$. Finally, we take the expectation of (27) and the inequality (26) follows immediately. □

We are now ready to prove Theorem 1.

Proof of Theorem 1.

First, we prove **part (a)** for the SBL-DMD algorithm. Note that

$$\sum_{t=1}^{T}\tau(x_t,a_t)=\sum_{t=1}^{T}r(x_t,a_t)+\sum_{j=1}^{K}\beta_j\left(n(j)-\alpha T\right)^+$$
$$\leq\sum_{t=1}^{T}r(x_t,a_t)+\sum_{j=1}^{K}\beta_j\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+ +\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+, \tag{29}$$

where the inequality follows from $(X+Y)^+\leq X^++Y^+$ for any $X,Y\in\mathbb{R}$. Putting the inequalities (25), (26), and (29) together, we obtain, for the exploitation rounds of the SBL-DMD algorithm,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}r(x_t,a_t)\right]\geq&\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]-\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+\right]-\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right]\\ \geq&\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]-O\left(\sqrt{KT\log T}\right)-\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right]\\ \geq&T\cdot\mathsf{OPT}-O\left(\sqrt{KT\log T}\right)-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)-\mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right]-\frac{2\eta}{\sigma}T-\frac{1}{\eta}D_\varphi(\lambda,\lambda^1)\\ =&T\cdot\mathsf{OPT}-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)-\mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right]-\frac{2\eta}{\sigma}T-\frac{1}{\eta}D_\varphi(\lambda,\lambda^1).\end{aligned}$$

Next, we relax the assumption that p_i is known to the algorithm for each i by demonstrating that observing $\hat{p}_i$ only will only incur an additional regret of an order lower than $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. We show that using the empirical probability $\hat{p}_i$ (instead of the true one p_i) only incurs an additional regret of an order lower than the one in Lemma 3(b), i.e., $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)+O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)$. Note that, by Assumption 2, the estimate satisfies $|\hat{c}_{ij}^t-c_{ij}^t|=O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ads with probability at least $1-t^{-3}$. Furthermore, the empirical distribution estimate $\hat{p}_i^t$ satisfies that with probability at least $1-t^{-4}$, for any context $i\in\mathcal{I}$, we have $|\hat{p}_i^t-p_i|\leq O(\sqrt{\log t/t})$ (see the discussions in Appendix B.2). Combining the above two error estimation bounds on $\hat{c}_{ij}^t$ and $\hat{p}_j$, we have, by the definitions of OPT^t and OPT,

$$\mathsf{OPT}^t\geq\mathsf{OPT}-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)-O\left(t^{-\frac{1}{2}}(\log t)^{\frac{1}{2}}\right)-O\left(t^{-\frac{5}{6}}(\log t)^{\frac{5}{6}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)=\mathsf{OPT}-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{30}$$

Hence, Lemma 3(b) and inequality (16) continue to hold if we replace p_i with $\hat{p}_i^t$ for each context i and each round t. The rest of Theorem 1's proof remains the same. Summarizing our argument above, we have shown that the expected regret of the SBL-DMD algorithm is bounded by

$$O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)+\mathbb{E}\left[\sum_{t=1}^{T}s_t(\lambda)\right]-\frac{2\eta}{\sigma}T-\frac{1}{\eta}D_\varphi(\lambda,\lambda^1),$$

i.e., part (b) holds.

Finally, we prove **part (a)** for the SBL-RS algorithm. We follow exactly the same argument as the proof for part (b) and derive, from inequalities (24), (26), and (29), that, under SBL-RS,

$$\mathbb{E}\left[\sum_{t=1}^{T}r(x_t,a_t)\right]\geq T\cdot\mathsf{OPT}-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$$

Together with (30), we have, for $\tau_m - \tau_{m-1} = 1$, the expected regret of the SBL-RS algorithm is bounded by

$$O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right),$$

i.e., part (a) holds if the algorithm re-solves the empirical dual in each period.

We next show that it suffices to solve the empirical dual problem with a fixed epoch of size $O(T^{\frac{2}{3}})$. Since the regret bound is of order $\tilde{O}(T^{\frac{2}{3}})$, we can discard the first $T^{\frac{2}{3}}$ periods without affecting the order of the regret bound. After the first $T^{\frac{2}{3}}$ periods, with a fixed epoch schedule such that $\tau_{m+1} - \tau_m = O(T^{\frac{2}{3}})$, we have $\tau_m \geq (1/2)\tau_{m+1}$. Therefore, at round t the additional regret it incurs to solve the empirical dual program is at most a constant multiplication of $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$, which is still of order $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. Therefore, summing this bound over t from 1 to T, we have that the total additional regret from setting $\tau_m - \tau_{m-1} = O(T^{\frac{2}{3}})$ is of order $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. This completes the proof of Theorem 1(a). □

B.4. Online Mirror Descent

To make our paper self-contained, we present a standard result on online mirror descent, which is inherited from Proposition 5 of Balseiro et al. (2020).

PROPOSITION 1 **(Online Mirror Descent)**. *With the sequence of convex functions* $s_t(\lambda) = -\sum_{j\in[K]\setminus a_t}\alpha\lambda_j + (\hat{c}^t_{i_t a_t} - \alpha)\lambda_{a_t}$, *let* $z_t \in \partial_\lambda s_t(\lambda)$ *be a subgradient and*

$$\lambda^{t+1} = \underset{0\leq\lambda_j\leq\beta_j,\forall j\in A}{\arg\min}\ \langle z_t, \lambda\rangle + \frac{1}{\eta}D_\varphi(\lambda, \lambda^t).$$

By the definition of $s_t(\lambda)$, *the subgradients are bounded by* $||z_t||_\infty \leq \alpha + 1 \leq 2$. *Suppose the reference function* φ *is* σ*-strongly convex with respect to* L_1*-norm. Then, for any* $\lambda_j \in [0, \beta_j]$ $(1 \leq j \leq K)$, *we have*

$$s^t(\lambda^t) - s^t(\lambda) \leq \frac{2\eta}{\sigma} + \frac{1}{\eta}D_\varphi(\lambda, \lambda^t) - \frac{1}{\eta}D_\varphi(\lambda, \lambda^{t+1}),$$

and

$$\sum_{t=1}^{T}[s^t(\lambda^t) - s^t(\lambda)] \leq \frac{2\eta}{\sigma}T + \frac{1}{\eta}D_\varphi(\lambda, \lambda^1).$$

The proof of Proposition 1 mainly follows from the first order conditions and the Three-Point Property of the Bregman projection. We refer interested readers to Proposition 5 in Balseiro et al. (2020) for proof details.

Appendix C: Additional Empirical Analysis

In this section, we present the following additional empirical analysis: (a) Validation for the causality of Figure 1; (b) randomization check for our field experiment; (c) verification of SUTVA for the experiment; and (d) regression analysis as robustness checks for the short-term impact of oSBL (i.e., Table 2).

C.1. Validation for the Causality of Figure 1

In this section, we casually estimate the effect of cold start performance on ad retention using two different methods. We first conduct a propensity score matching (PSM) analysis using pre-experiment data. Specifically, we access the data of all new ads created between May 1, 2020 and May 7, 2020, and examine their performance before our two-sided experiment which started on May 23, 2020. Specifically, we use PSM to construct the treatment and control groups of ads. The treatment (resp. control) group of ads correspond

to those ads whose conversions is greater or equal to (resp. below) 10 during their cold start period. We include all potential confounding variables we are aware of, such as bidding price, budget, industry, and target strategy, to match the control sample with the treatment sample using logistic regression. Note here the target strategy is set by the advertiser before the ad campaign, which (based on age, gender, location, phone brand and so on) chooses a subset of eligible platform users for displaying ads.

There are in total 97,273 new ads created between May 1, 2020 and May 7, 2020. We construct a subsample of 22,994 new ads with PSM. We then run a linear regression with the indicator variable for obtaining at least 10 conversions as the treatment variable and controlling for other features used in matching, i.e.,

$$\text{PSM}: y_j = \beta_0 + \beta_1 \text{Treatment}_j + X_j + \epsilon_j,$$

where y_j corresponds to whether ad j is retained in the two week course after its cold-start period, Treatment_j corresponds to whether ad j is the treatment group, and X_j are the features. Our regression result shows that if a new ad gains more than 10 conversions during the cold start period, the retention rate is significantly increased by 15.03% (p-value<0.0001), which is reported in the column (2) of Table 5. Thus, we have partially established the causality for Figure 1 that reaching the cold-start conversion threshold during the first few days could significantly boost the retention rate of an ad. We also report the naive regression result with PSM, i.e.,

$$\text{Linear Regression}: y_j = \beta_0 + \beta_1 \text{Treatment}_j + \epsilon_j,$$

in column (1) of Table 5 for comparison. Note that this corresponds to a straightforward t-test.

One may still worry whether some other unobservable confounding variables may invalidate the above matching-based result. To further justify the validity of Figure 1, we leverage our experimental data to casually estimate the effect of cold-start success on ad retention. Specifically, we use our two-sided experiment as an instrumental variable (IV) to identify the effect of whether a new ad gaining more than 10 conversions during the cold start period on its retention (encoded as the binary variable representing whether the ad remains active on the platform for everyday in the following two weeks after the cold start period). We adopt the two-stage least squares (2SLS) specification given by (31).

$$\begin{aligned} &\text{IV-First Stage: } s_j = \alpha_0 + \alpha_1 \text{Treatment}_j + X_j + \epsilon_j \\ &\text{IV-Second Stage: } y_j = \beta_0 + \beta_1 \hat{s}_j + X_j + \epsilon_j \end{aligned} \tag{31}$$

To estimate the impact of cold start success on ad retention, we denote $s_j = 1$ if ad j gains more than 10 conversions during the experiment; otherwise $s_j = 0$. X_j is ad-specific features, such as bidding prices, budget, industry, and target strategy, for ad j. $\text{Treatment}_j = 1$ if ad j is in the treatment group; otherwise $\text{Treatment}_j = 0$. We use $y_j = 1$ to denote that ad j stays on the platform in the next two weeks after the cold-start period; otherwise $y_j = 0$. We remark that Treatment is a valid instrument in this setting. On the one hand, the p-values of the weak instrument tests are smaller than 10^{-5} so the strong first-stage assumption holds. On the other hand, it seems unlikely that our experiment could impact the retention of an ad through a channel other than conversions, so exclusion restriction also holds.

Under the 2SLS specification (31), we further validate the causality of Figure 1 and demonstrate that gaining 10 conversions during the cold start period will significantly increase the ad retention rate by 15.20% (p-value is less than 0.0001), which is reported in the column (3) of Table 5. Note this result effectively matches that in the column (2).

Table 5 Effect of Cold Start Success on Ad Retention

	Methodology:		
	Benchmark	Matching	Instrumental Variable
	(1)	(2)	(3)
Absolute Effect Size	17.01%****	15.03%****	15.20%****
Standard Error	(0.004)	(0.006)	0.003
Experimental Data	No	No	Yes
Observations	97,273	22,994	49,544
Industry Fixed Effects	No	Yes	Yes
Bidding Price	No	Yes	Yes
Budget	No	Yes	Yes
Target Strategy	No	Yes	Yes

Note: *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001.

Table 6 Randomization Check of the Experiment

Panel A: Randomization Check on the Ad side				
		Treatment ads	Control ads	p-value of t-test
Statistics during the Experiment	Number of New Ads	34,605	34,076	
	Bidding Price	48.14	48.17	0.91
		(52.24)	(51.45)	
	Proportion of Ads for iOS Users	24.1%	24.2%	0.98
		(0.427)	(0.428)	
	Proportion of Ads for UI Version X	30.3%	28.3%	0.69
		(0.459)	(0.450)	
	Proportion of Ads in Game Industry	13.8%	13.7%	0.98
		(0.086)	(0.081)	
	Proportion of Ads in Education Industry	0.75%	0.67%	0.93
		(0.086)	(0.082)	
	Proportion of Ads in Finance Industry	1.75%	1.87%	0.93
		(0.131)	(0.135)	
Panel B: Randomization Check on the UV side				
		Treatment UV	Control UV	p-value of t-test
Statistics during the Experiment	Number of Users	197,460,792	197,401,621	
	Male Proportion	0.540	0.540	>0.99
		(0.491)	(0.491)	
	Average Revenue per User	0.95	0.95	>0.99
		(27.15)	(27.14)	
	Average Impressions per User	23.36	23.24	0.95
		(17900)	(17864)	
	Average Clicks per User	3.195	3.20	0.99
		(4455)	(4458)	
	Average Conversions per User	0.041	0.040	0.88
		(32.80)	(32.25)	

Note: Standard deviations in Panel A are clustered at the ad level and reported in the parentheses. Standard deviations in Panel B are clustered at the user level and reported in the parentheses. To protect sensitive data, the reported metrics are rescaled.

C.2. Randomization Check of the Field Experiment

To confirm the success of our randomization in the two-sided experiment, we check the randomization on both the ad side and the UV side before oSBL coming into effect. For the ad side randomization check, we report the ad side randomization check results in Table 6 Panel A, where the numbers are re-scaled to protect the sensitive data. Table 6 Panel A shows that treatment and control ads in our sample were similar in bidding prices, the proportion of ads targeting iOS users, the proportion of ads targeting UI Version X, and the proportion of ads in various industries. We remark that these features are all submitted by the advertiser once s/he launches a new ad on the DSP and, therefore, are not affected by the algorithm of choice. Similarly, the UV side randomization check results are reported in Table 6 Panel B, where the numbers are also rescaled to protect the sensitive data. As we can see from Table 6 Panel B, treatment UVs and control

UVs generate similar revenues, ad impressions, ad clicks, and ad conversions per hour. The proportion of female users in the treatment group is also similar to that in the control group. We have thus confirmed that the treatment ads (resp. UVs) and control ads (resp. UVs) in our sample are comparable, implying that any difference between groups after the experiment started should be attributed to whether our new oSBL algorithm has been implemented.

C.3. Verification of SUTVA for the Experiment

To verify SUTVA for our two-sided experiment, we examine two additional assumptions during our experiment time period: (a) The CTR and CVR distributions of the mature ads are not affected by the cold start algorithms applied to different UVs; and (b) The total number of ad impressions displayed to a user is not affected by the cold start algorithms applied to different ads. To test the first assumption, we sample 13,337 mature ads one day before the experiment and compare their empirical CTR before and during the experiment. The average CTR before the experiment is 13.11% with standard deviation 0.099, while the average CTR during the experiment is 13.19% with standard deviation 0.100. The p-value of the pairwise t-test is 0.284 and 0.481 for CTR and CVR, respectively, implying that our algorithm does not substantially change the CTR and CVR of mature ads *during* the experiment. To test the second assumption, we conduct a t-test of average ad impressions per user for the treatment and control ad impressions in our experiment. We find that the p-value is 0.96. Hence, our algorithm does not change the number of ad impressions significantly. Therefore, SUTVA holds for our two-sided experiment.

C.4. Robustness Check for Regression-Based Results

In this subsection, we replicate our main results for the short-term impact of oSBL (i.e., Table 2) using the following linear regression specification which controls for the ad features to improve the efficiency of our estimators:

$$\text{Performance Indicator}_j = \alpha_0 + \alpha_1 \text{Treatment}_j + X_j + \epsilon \tag{32}$$

For the impact of the new algorithm on cold start reward and cold start success rate, Treatment_j is 1 if ad j is in the treatment group, otherwise 0; X_j is ad-specific features including the industry category (of the advertiser), bidding price, budget, and the to target strategy. The target strategy means that advertisers can predetermine whom to display the ad based on users' age, gender, location, phone, device features, and so on. We use the specification (32) to check the robustness of results on the revenue implications of our oSBL algorithm, where Treatment_j is 1 if ad j is in the treatment group, otherwise 0; For conciseness, we only report the three most important metrics of the platform, namely the cold start success rate and the cold start reward, as well as the revenue. The result of specification (32) is presented in Table 7. The regression-based results indicate that, after controlling for ad-specific characteristics, our algorithm can significantly increase the cold start success rate by 41.22% and the cold start reward by 53.87%, which are similar to the model-free results on the short-term impact of oSBL (see Section 6.1) in both directions and magnitudes.

Table 7 Regression-Based Effects of the Algorithm

	Dependent Variable:			
	Cold Start Reward	Cold Start Success Rate	Revenue Per User	Objective Value
	(1)	(2)	(3)	(4)
Treatment−Control	73.0	0.0146	-0.0072	556,607
	(5.04)	(<0.001)	(<0.001)	
Relative Effect Size	41.22%****	53.87%****	-0.592%**	0.157%****
Model-Free Relative Effect Size	47.71%****	61.62%****	-0.717%**	0.147%****
Industry Fixed Effects	Yes	Yes	Yes	Yes
Bidding Price	Yes	Yes	Yes	Yes
Budget	Yes	Yes	Yes	Yes
Target Strategy	Yes	Yes	Yes	Yes

Note: *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001. Standard errors in column (1) and (2) are at the ad level and reported in the parentheses. Standard error in column (3) is at the user level.

Appendix D: Additional Simulation Analysis

In this section, we present additional simulation analysis to complement our theoretical and empirical results. In Appendix D.1, we quantify the estimation biases under single-sided experiments (see also Section 5). Appendix D.2 numerically illustrates the expected regret of our algorithm (Theorem 1). Appendix D.3 numerically shows that for a wide range of cold start reward parameter β_j, our SBL algorithm can successfully boost the long-term total advertising revenue of the platform.

D.1. Estimation Biases with One-Sided Experiments

To begin with, we describe the simulation model based on the real data of Platform O. The simulation studies presented in Appendices D.1 and D.2 are based on this model.

Data Set. Our data set contains one-week log records of 300 ads, which consists of three types of data from December 1, 2019 to December 7, 2019, including (1) impression, click and conversion records of the ads, (2) the user page-view records with gender, location, age, device features of each user displayed with the ads, and (3) the bidding prices of each ad. However, to protect the platform's sensitive data, all the reported values in this section are re-scaled.

Advertising Mechanism. To simplify the advertising mechanism of our simulator, we only consider the CPC billing option and first-price auction in our simulation. Thus, we ignore the conversions and focus on clicks. We generate the incoming users' ad view requests with features using the shuffled real-world user page-view records. Each ad joins the auction with the true bidding price and the pCTR generated by a logistic regression model described below. After allocating an ad to the user page view, the simulator gets the feedback whether the ad is clicked based on the ground truth CTR and a binomial click probability. If the ad is clicked, the advertiser is charged the bidding price for each click. Meanwhile, the pCTR model is updated with this new feedback data of user click.

Ground Truth CTR Model and pCTR Model. The simulator is equipped with a ground truth click-through model for each ad via fitting a logistic regression model on its historical impression and click records. The pCTR model is essential in deciding which ad will win each user impression. In our simulator, the pCTR model is built upon a logistic regression model with randomly initialized parameters, which are updated in real-time based on the click-through feedback.

To mimic the real-world advertising platform where mature ads and new ads are mixed together, we randomly classify the 300 ads in the simulation into 100 new ads and 200 mature ads. The pCTR models

Table 8 Bias Analysis of Different Estimators

	Ad-Side Experiment		UV-Side Experiment		Two-Sided Experiment	
	Treatment	Control	Treatment	Control	Treatment	Control
Cold-Start Success Rate	0.068	0.024	0.050	0.038	0.060	0.038
	(0.011)	(0.017)	(0.007)	(0.013)	(0.007)	(0.008)
Value of Estimator	4.4%**		1.2%*		2.2%**	
Bias/Global Treatment Effect	120%		-40%		10%	

Note: *p<0.05; **p<0.01; ***p<0.001; ****p<0.0001. Standard errors are reported in the parentheses.

of mature ads are pre-trained before the start of simulation. The simulation contains two phases, the cold start phase and the long-term stationary phase. During the cold-start phase, there are in total 1,000,000 user page-views to be allocated. After the cold start phase, consistent with Figure 1, we assume that a new ad with fewer accumulated clicks have a higher probability to leave the advertising platform. The new ads who stay on the platform together with the 200 mature ads proceed to the stationary phase with another 1,000,000 user page-views.

We separate 2,000,000 page views equally into the cold start phase and the stationary phase. We capture ads' retention behavior after the cold start phase (see Figure 1), using a piecewise linear model to fit the real data with the following specification:

$$\mathbb{P}[\mathsf{retention}] = \gamma_1 + \gamma_2 \times \min\left\{1, \#\mathsf{clicks}/\alpha T\right\},$$

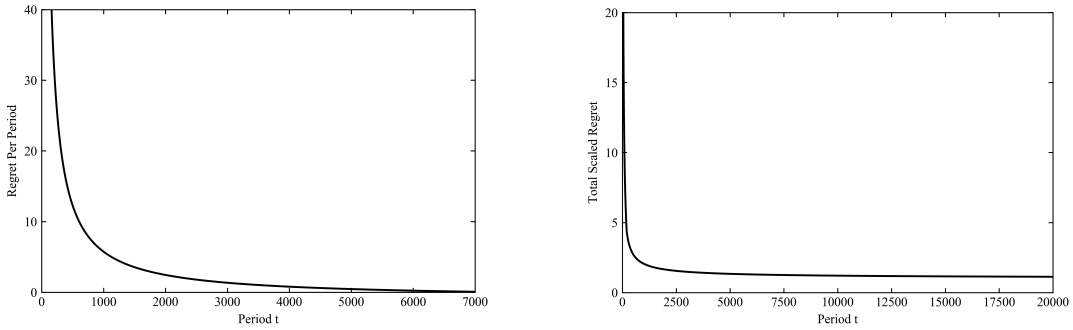
where $\#\mathsf{clicks}$ is the number of accumulated clicks during the cold start phase, $\alpha T = 1000$ is the target number of clicks with $\alpha = 0.001$ and $T = 1,000,000$. To protect the sensitive data, we are required not to report the coefficients γ_1 and γ_2.

To illustrate the potential estimation biases induced by the violation of SUTVA with Ad-side and UV-side experiments, we run three numerical simulations (Ad-side randomization experiment (see Figure 4(a), UV-side randomization experiment (see Figure 4(b), and two-sided experiment (see Figure 5) in our simulation system with 100 new ads and 200 mature ads, as well as 1,000,000 user views. We fix $\alpha = 0.001$ and $\beta = 2b$ for all experiments in this set of simulations. In all these simulations, 50% of UVs/Ads are randomly assigned into the treatment condition, and the other 50% into the control condition.

We replicate the simulation for each randomized experiment for five times and report corresponding estimation results of cold start success rate in Table 8. Our simulation results demonstrate that the ad-side experiment significantly overestimates the treatment effect of the proposed algorithm, whereas the user-side experiment underestimates the effect. Furthermore, the two-sided equips us with an unbiased estimate. Therefore, our simulation results necessitate and validate our two-sided experiment design by showing that whereas one-sided experiments are likely to produce substantially biased estimates, our novel two-sided design helps correct such biases.

D.2. Regret Illustration

To numerically illustrate the regret of our algorithm, we consider a pure cold start setting with new ads only. We randomly select 100 new ads from the data set as well as 20,000 impressions to be allocated. The target number of clicks for each ad per one period is $\alpha = 0.001$, which is scaled in consistency with the cold start success criteria of 10 conversions in the three-day cold start horizon. We set the cold start reward coefficient



(a) Average Regret (b) Average Regret Scaled by $t^{-2/3}(\log t)^{-1/3}$

Figure 8 Average Regret and Scaled Regret in the Simulation with $\alpha = 0.001$ and $\beta_j = 2b_j$

$\beta_j = 2b_j$ for each ad j. We run the simulation for 50 times and compute the average regret. The results are plotted in in Figure 8, which confirms our theoretical result (Theorem 1) that the regret of SBL algorithms is bounded by $O(t^{2/3}(\log t)^{1/3})$.

D.3. Impact of the Cold Start Reward Coefficient on Long-Term Revenue

As shown in Section 5.2, the online implementation of our algorithm sets the cold start reward coefficient at $\beta_j = 2b_j$ for each ad j. In this subsection, we leverage the simulation model built in Section 6.3 to demonstrate that for a wide range of choices of β_j, our SBL algorithm can successfully boost the long-term total advertising revenue of the platform. Similar to the simulation setting in Section 6.3, to estimate the global treatment effect of the cold start reward coefficient β/b on long-term revenue, we use the data with 12 million impressions from those during April 9, 2020 and April 30, 2020. The specific simulation setting and the model validation has been documented in Section 6.3, we randomly sample 1.2 million impressions with replacement for each simulation and replicate 10 times via Bootstrap to estimate the long-term revenue of the oSBL. All the following results pass the t-test with p-value smaller than 10^{-3}.

In this subsection, however, we emphasize on the robustness of the choice of cold start coefficient β/b, which boosts the positive long term revenue within a wide range. In the regard, we conduct a sensitivity analysis with three varying parameters in the simulation. Δ_r, which refers to the average relative increase of the retention length for mature ads, Δ_{CTR}, which refers to the average relative increase of the CTR for mature ads, and the cold start reward coefficient β/b. To obtain a complete picture on the global treatment effect of our algorithm, we conduct a sensitivity analysis with our simulation model by varying $\Delta_r \in \{1\%, 2\%, 3\%\}$, varying Δ_{CTR} from 0 to 15%, and varying β/b from 0 to 3, assuming that oSBL is applied to all ads and UVs. The baseline revenue is denoted by R_0 and the revenue under the oSBL algorithm denoted by $R(\Delta_r, \Delta_{CTR}, \beta/b)$ (so $R_0 = R(0,0)$). We are interested in the relative advertising revenue increase associated with oSBL:

$$\Xi(\Delta_r, \Delta_{CTR}) = \frac{R(\Delta_r, \Delta_{CTR}, \beta/b) - R_0}{R_0} \times 100\%$$

The results of the sensitivity analysis presented in Figures 9, 10, and 11 demonstrate that for a wide range of Δ_r and Δ_{CTR}, our algorithm oSBL can successfully boost the long term revenue with a flexible choice of β/b.

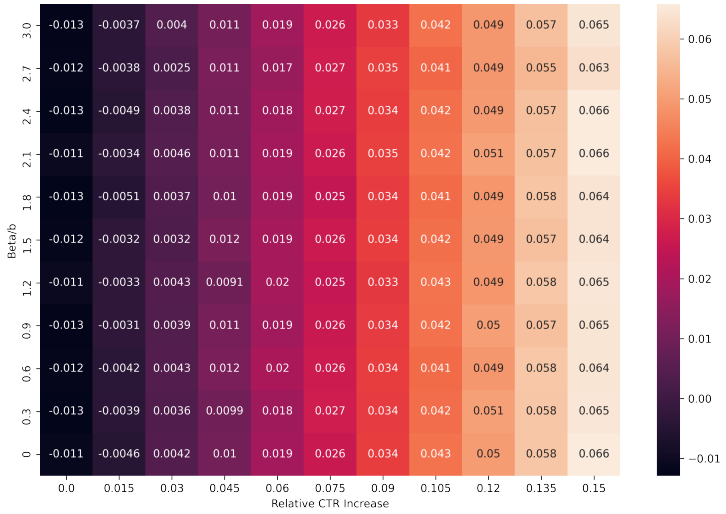


Figure 9 **Global Treatment Effect of oSBL on Advertising Revenue with $\Delta_r = 0.01$**

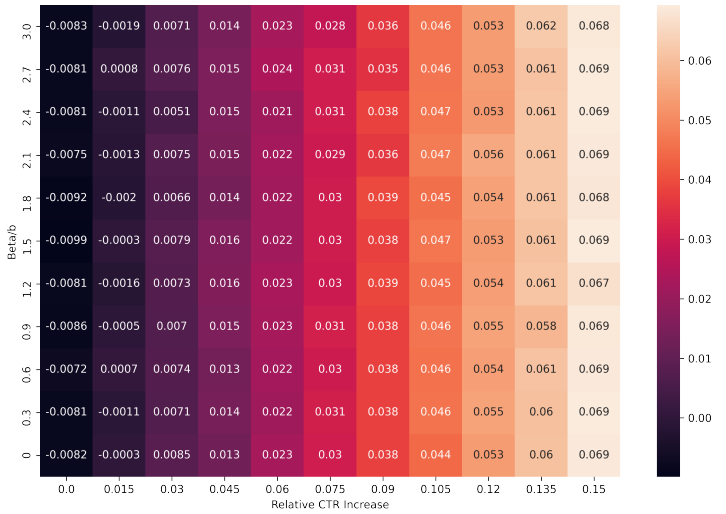


Figure 10 **Global Treatment Effect of oSBL on Advertising Revenue with $\Delta_r = 0.02$**

Appendix E: Performance of Subgradient Descent Algorithm for Solving Duals

To demonstrate the effectiveness of subgradient descent to obtain the shadow bids λ for ad allocation, we compare our shadow-bidding-price-based ad allocation, where λ is computed by subgradient descent method with (a) the Simplex method which solves the primal directly, (b) another gradient-based method SHALE (Bharadwaj et al. 2012, Hojjat et al. 2017), and (c) the current practice of Platform O, namely showing the ad with maximum eCPM without considering the cold start reward. We examine a small-scale instance with 100 Ads and 10,000 UVs in an offline setting. However, our online implementation solves the dual instances

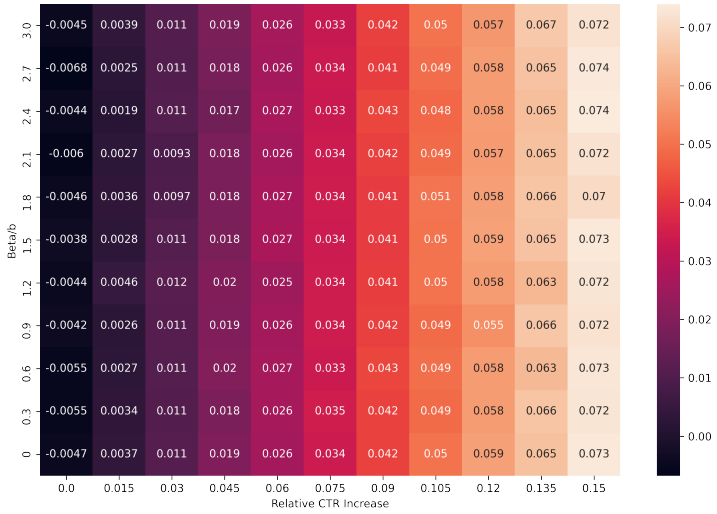


Figure 11 Global Treatment Effect of oSBL on Advertising Revenue with $\Delta_r = 0.03$

with more than 10,000,000 UVs, which is impossible to solve in a reasonable time by the standard Simplex approach. The stopping condition of our subgradient descent algorithm is when the duality gap is less than $O(10^{-4})$, which is consistent with our online implementation. Other parameters such as bidding prices b_j and pCTRs are directly from the real data. The computational results are summarized in Table 9.

Table 9 Objective Value Comparison

	Current Practice (1)	SHALE (2)	Subgradient Descent (3)	Simplex (4)
Revenue	288,556	284,913	278,598	278,588
Cold Start Reward	67,747	76,171	98,429	98,679
Total Objective Value	356,303	361,084	377,026	377,267

The objective value is the sum of the revenue and cold start reward. The relative difference between our dual-based subgradient descent approach and the optimal objective value is less than 0.07%, which suggests that the gap induced by integer round-offs and the stopping condition in our algorithm is negligible. Moreover, our algorithm performs substantially better than the SHALE algorithm (Bharadwaj et al. 2012).

Appendix F: Robustness Check of the UV Sampling Rate in oSBL

In this section, we conduct robustness check for the UV sampling rate, which shows that even a low sampling rate of 1% for user views could already cover most of the new ads and produce robust dual solutions. Moreover, considering that both memory and computational time increase linearly with the sampling rate, we choose 4% sampling rate for our online implementation, which strikes a good balance of sample representativeness and computational time.

Table 10 Robustness Check of the UV Sampling Rate

	Sampling Rate of UV r		
	$r=0.04$	$r=0.02$	$r=0.01$
	(1)	(2)	(3)
The number of new Ads	6216	6216	6216
Mean of λ	64.21	64.22	64.25
Standard deviation of λ	62.96	62.96	63.05
25^{th} *percentile of* λ	15.00	15.00	15.00
50^{th} *percentile of* λ	57.60	57.60	57.21
75^{th} *percentile of* λ	90.00	90.00	90.00

Note: The differences between λ's calculated by different sampling rates are not significant. P-values of t-tests between (1) and (2), (1) and (3), and (2) and (3) are, respectively, 0.716, 0.155, and 0.280.

Appendix G: Training with Neural Networks to Predict CTR

In this section, we show that the there exists a fully connected neural network satisfying *Prediction Oracle* with high probability (i.e., Assumption 2 holds) under either (a) the lazy training regime or (b) the training algorithm with gradient descent.

Before presenting the formal results and their proofs, we first introduce the fully connected neural network and its initialization procedure. In recent years, deep-learning-based recommender systems are flourishing and widely used in practice (see the review paper Zhang et al. 2019). A large-scale DSP like Platform O is also armed with deep neural networks to predict the CTR and CVR of ads. In practice, due to the limited computational resource and high requirement on fast response, the "funnel" structure is widely adopted. For example, YouTube's recommender system (see Covington et al. 2016) uses a rough deep learning model which is very efficient but less accurate, to select hundreds out of millions of videos. Then, it uses a more sophisticated deep-learning-based ranking model with more feature inputs to choose dozens of videos from the hundreds selected in the previous step. Platform O and other video sharing platforms adopt a similar recommendation strategy. Specifically, for Platform O's DSP, there are two stages before an ad enters the final auction: filtering and pre-ranking, both of which adopt rough deep neural network models to rule out the ads not suitable for the user impression. Then, at the final auction stage, Platform O uses a set of fully connected neural networks with the ReLU activation function, i.e., $\sigma(\cdot)=\max\{\cdot,0\}$, to predict the CTR and CVR. Since there are only around 150 ads joining the auction, Platform O typically uses an individualized neural network for predicting the CTR of each ad rather than a unified model for all ads. Without loss of generality, we assume all hidden layers of the neural network have the same number of nodes. And we denote $L\geq 2$ as the network depth, w as the number of nodes in each hidden layer, and w_0 as the dimension of the context/feature vector, i.e., $x_{ij}\in\mathbb{R}^{w_0}$ for all $i\in\mathcal{I},\ j\in A$. Following the convention of the neural network literature (e.g., Cao and Gu 2019, Chizat et al. 2019), we parameterize the neural network by $\theta\in\mathbb{R}^d$, where the effective dimension $d=w^2(L-2)+ww_0+w$. Then, we can use the function $H_j(x_{ij},\theta)=\sqrt{w}W_L\sigma(W_{L-1}\sigma(\cdots\sigma(W_1x_{ij})))$ to represent the output of the neural network given the parameter θ, for any ad j and context i, where $\theta=[\mathrm{vec}(W_1),\cdots,\mathrm{vec}(W_L)]$, $W_1\in\mathbb{R}^{w_0\times w}$, $W_i\in\mathbb{R}^{w\times w}$, $2\leq i\leq L-1$, $W_L\in\mathbb{R}^{w\times 1}$ and the operator $\mathrm{vec}(\cdot)$ refers to representing the matrix as a vector.

In practice, the initialization procedure may take into account the domain knowledge of the context. In our analysis that follows, we adopt the initialization procedure in He et al. (2015), known as the *He Initialization*

to set θ_0. For each layer $1 \le l \le L-1$, we set W_l to be $\begin{pmatrix} W & 0 \\ 0 & W \end{pmatrix}$, where each entry of this matrix W is randomly and independently drawn from a normal distribution $N(0, 2/w)$. The parameter of the last layer is initialized as $(w^T, -w^T)$ where each entry of vector w is randomly and independently drawn from distribution $N(0, 1/w)$. One can verify that under this initialization procedure of θ_0, it holds that $H_j(x_{ij}, \theta_0) = 0$ for all context i and ad j (see Cao and Gu 2019, He et al. 2015).

To validate Assumption 2 for fully connected neural networks, we make additional technical assumptions as follows, which are mild and commonly made in the related literature (e.g., Cao and Gu 2019, Zhou et al. 2020).

ASSUMPTION 3 **(Finite and nonparallel Contexts)**. *(a) The number of context type i is finite, i.e., the cardinality of the context set $|X| = m < +\infty$, and m is bounded by a fixed polynomial of T, i.e., $m \le O(T^k)$ for a some k. (b) For any pair of contexts $x_{ij}, x_{i'j'} \in X$ $(i \ne i'$ or $j \ne j')$, x_{ij} and $x_{i'j'}$ are not parallel. (c) The L_2-norm of each context is normalized to 1, i.e., $\|x_{ij}\|_2 = 1$ for any context $i \in [m]$ and ad $j \in A$. (d) The j^{th} component of x is equal to the $(j + d/2)^{th}$ component for any context $x \in X$ and $j \le d/2$*

The parts (a) and (b) of Assumption 3 are mild, while parts (c) and (d) are just for the convenience of analysis. Notice that part (d) can always be satisfied by transforming any context x to a new one $x' = [x, x]/\sqrt{2}$.

G.1. Lazy Training Regime

The recent progress of *Neural Tangent Kernels* (e.g., Cao and Gu 2019, Jacot et al. 2018, Arora et al. 2019, Zhou et al. 2020) theoretically characterizes the representation power of a neural network. Following this literature, we use $\boldsymbol{H}$ to denote the Neural Tangent Kernel Matrix in the same way as Definition 4.1 of Zhou et al. (2020). As discussed in Zhou et al. (2020) Assumption 4.2, $\boldsymbol{H} \succeq \gamma I$ always holds for some $\gamma > 0$, where I is the identity matrix under Assumption 3. This ensures that the Neural Tangent Kernel Matrix $\boldsymbol{H}$ is always nonsingular. Lemma 5 below shows that, as long as the ground truth CTR can be represented as a bounded function of user contexts and ads, then the fully connected neural network with a large width w can accurately predict this CTR with high probability in terms of the *He Initialization*.

LEMMA 5 **(Lemma 5.1 in Zhou et al. (2020))**. *Under Assumption 3, for any $j \in A$ there exists a constant $C > 0$ such that, if $w \ge Cm^4L^6 \log(m^2L/\delta)/\gamma^4$, then with probability $1-\delta$ over the He Initialization of the parameter θ_0, there exists a $\theta^* \in \mathbb{R}^d$ such that, for any $i \in \mathcal{I}$,*

$$c_{ij} = \langle \nabla_\theta H_j(x_{ij}, \theta_0), \theta^* - \theta_0 \rangle, \sqrt{w}\, \|\theta^* - \theta_0\|_2 \le \sqrt{2c^T \boldsymbol{H} c},$$

where $c := [c_{ij}]_{i \in \mathcal{I}} \in \mathbb{R}^m$.

Notice that the approximation of ground truth CTR in Lemma 5 is linear in the gradient $\nabla_\theta H_j(x_{ij}, \theta_0)$ parametrized by $\theta^* - \theta_0$. The original neural network for the ad j mapping $H_j(\cdot, \cdot)$ is now divided into two steps. First, it maps the context x_{ij} to the gradient $\nabla_\theta H_j(x_{ij}, \theta_0)$. This is a static mapping that depends on the initialization θ_0, but independent of the parameter θ. The second step linearly maps the gradient $\nabla_\theta H_j(x_{ij}, \theta_0)$ to the true CTR, c_{ij}. As a consequence, to train the neural network under this regime, it

suffices to fit a linear function parameterized by θ. This training method is referred to as *Lazy Training* in the literature (Chizat et al. 2019). Specifically, lazy training with the regularized least square loss function at round t for ad j is equivalent to solving the following minimization problem:

$$\min_{\theta} w\lambda_0\|\theta-\theta_0\|^2+\sum_{\tau\in\mathcal{T}_j^t}(z_\tau-\langle\nabla_\theta H_j(x_{i_\tau a_\tau},\theta_0),\theta-\theta_0\rangle)^2, \tag{33}$$

where $\mathcal{T}_j^t$ denotes the time periods until round t in which ad j is played, x_{i_τ,a_τ} represents the context vector associated with context i_τ and ad a_τ realized at round τ. λ_0 is the regularized parameter, z_τ denotes the corresponding click-through outcome of round τ. With lazy training, we effectively linearize the neural network for CTR prediction, thus reducing it to a linear regression model. Lazy training facilitates us to focus on cold start algorithm design, without delving into the details of how a neural network shall be trained. Similar approaches, usually referred to as *Kernelized Contextual Bandits*, have been adopted in the contextual bandit literature (e.g., Valko et al. 2013). As identified by Chizat et al. (2019), the lazy training phenomenon, where a neural network behaves similarly to a linear model when the parameter θ is close to the initialization parameter θ_0, will occur when the neural network is over-parameterized. In addition, Chizat et al. (2019) also show that the gradient flows of the lazy training process and the gradient descent training process (see Appendix G.2) are close to each other for over-parameterized neural networks. We also remark that the real online training procedure of Platform O's CTR/CVR prediction model is neither pure supervised learning nor lazy training, but a substantial compromise under limited computational resources. In this regard, incorporating the exact online training process into our regret analysis is unnecessary and beyond the scope of this paper. Although Chizat et al. (2019) empirically shows that lazy training might not perform well in some cases with biased gradients, this training method still provides a good theoretical understanding of how the CTR/CVR estimate is produced by neutral networks, and inspires us to validate Assumption 2 for neural networks with gradient descent training (see Appendix G.2). We are now ready to validate Assumption 2 for neural networks under the lazy training regime.

PROPOSITION 2 **(Prediction Oracle with Lazy Training)**. *Under Assumption 3, for any ad $j\in A$ with the prediction model (33) trained on n_j^t i.i.d samples drawn from $\mathcal{D}_X$ and the corresponding click-through outcome before round t, then we have that for any δ there exists a constant $C>0$, such that, if $w\geq Cm^4L^6\log(m^2L/\delta)/\gamma^4$ and $n_j^t\geq\Omega(d\log(dT))$, it holds that, with probability at least $1-\delta-T^{-4}$, for any context $i\in\mathcal{I}$ the following inequality holds:*

$$|\hat{c}_{ij}^t-c_{ij}|\leq O\left(\sqrt{\frac{d\log T}{n_j^t}}\right).$$

where $\hat{c}_{ij}^t$ is the predicted CTR at round t via model (33) with $0<\lambda_0\leq O(\sqrt{1/(2wc^T\boldsymbol{H}c)})$.

Proof of Proposition 2.

We first introduce some definitions. We use I_d to denote the identity matrix with dimension d. We define

$g_{ij} := \nabla_\theta H_j(x_{ij}, \theta_0)$, for all $i \in \mathcal{I}$ and $j \in A$. Following the standard lazy training with regularized squared loss (33), we can compute θ^t in closed form at each round t as follows:

$$\begin{aligned} A_j^t &:= w\lambda_0 I_d + \sum_{\tau \in \mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T, \quad & D_j^t &:= [g_{i_\tau a_\tau}^T]_{\tau \in \mathcal{T}_j^t} \\ b_j^t &:= \sum_{\tau \in \mathcal{T}_j^t} v_{i_\tau a_\tau} g_{i_\tau a_\tau}, & V_j^t &:= [v_{i_\tau a_\tau}]_{\tau \in \mathcal{T}_j^t} \\ \theta^t &:= (A_j^t)^{-1} b_j^t + \theta_0, & s_{ij}^t &:= \sqrt{g_{ij}^T (A_j^t)^{-1} g_{ij}} \end{aligned}$$

By Lemma 5, we consider the case, with high probability $1-\delta$, where CTR c_{ij} can be perfectly predicted via a linear mapping. Thus, at round t after observing n_j^t i.i.d samples, we have for any realized context $i \in \mathcal{I}$ at round t, with probability at least $1-\delta$:

$$\begin{aligned} |\hat{c}_{ij}^t - c_{ij}| &= |g_{ij}^T(\theta^t - \theta_0) - g_{ij}^T(\theta^* - \theta_0)| \\ &= |g_{ij}^T (A_j^t)^{-1} b_j^t - g_{ij}^T (A_j^t)^{-1} (w\lambda_0 I_d + (D_j^t)^T D_j^t)(\theta^* - \theta_0)| \\ &= |g_{ij}^T (A_j^t)^{-1} (D_j^t)^T (V_j^t - D_j^t(\theta^* - \theta_0)) - w\lambda_0 g_{ij}^T (A_j^t)^{-1} (\theta^* - \theta_0)| \\ &\le |g_{ij}^T (A_j^t)^{-1} (D_j^t)^T (V_j^t - D_j^t(\theta^* - \theta_0))| + w\lambda_0 M \left\| (A_j^t)^{-1} g_{ij} \right\|_2, \end{aligned} \tag{34}$$

where the first equality follows from the lazy training process $\hat{c}_{ij}^t = g_{ij}^T(\theta^t - \theta_0)$ (by Lemma 5), and the second from the identity $A_j^t = w\lambda_0 I_d + (D_j^t)^T D_j^t$ and $b_j^t = (D_j^t)^T V_j^t$. The inequality of (34) follows from the triangular inequality and $\|\theta^* - \theta_0\|_2 \le M$ (by Lemma 5, we take the value of M at $M = \sqrt{2c^T \boldsymbol{H} c / w}$). Because $\mathbb{E}[V_j^t - D_j^t(\theta^* - \theta_0)] = 0$, Azuma–Hoeffding inequality implies the following concentration inequality on the first term of (34).

$$\begin{aligned} \mathbb{P}\left[|g_{ij}^T (A_j^t)^{-1} (D_j^t)^T (V_j^t - D_j^t(\theta^* - \theta_0))| \ge \sqrt{\frac{1}{2}\log\frac{2}{\Delta}} s_{ij}^t \right] &\le 2\exp\left(- \frac{\log(2/\Delta)(s_{ij}^t)^2}{\left\| D_j^t (A_j^t)^{-1} g_{ij} \right\|_2^2} \right) \\ &\le 2\exp(-\log(2/\Delta)) = \Delta, \end{aligned} \tag{35}$$

where the second inequality follows from

$$\begin{aligned} \left\| D_j^t (A_j^t)^{-1} g_{ij} \right\|_2^2 &= (D_j^t (A_j^t)^{-1} g_{ij})^T D_j^t (A_j^t)^{-1} g_{ij} \\ &\le g_{ij}^T (A_j^t)^{-1} (I_d + (D_j^t)^T D_j^t)(A_j^t)^{-1} g_{ij} \\ &= g_{ij}^T (A_j^t)^{-1} g_{ij} = (s_{ij}^t)^2 \end{aligned} \tag{36}$$

Similarly, we have the bound $\left\| (A_j^t)^{-1} g_{ij} \right\|_2 \le s_{ij}^t$. Combining the above two inequalities (35) and (36), we have, with probability at least $1-\Delta$, $|\hat{c}_{ij}^t - c_{ij}| \le \left(w\lambda_0 M + \sqrt{\frac{1}{2}\log\frac{2}{\Delta}} \right) s_{ij}^t$. Notice that the gradient satisfies that $\|g_{ij}\|_2 \le \sqrt{wL}$ for all i and j (see Cao and Gu 2019), and the regularization parameter satisfies $\lambda_0 \le O(\sqrt{1/(2wc^T \boldsymbol{H} c)})$. Therefore, the regularization term satisfies

$$w\lambda_0 M \left\| (A_j^t)^{-1} g_{ij} \right\|_2 \le w \cdot O(\sqrt{1/(2wc^T \boldsymbol{H} c)}) \cdot \sqrt{2c^T \boldsymbol{H} c / w} \cdot s_{ij}^t = O(s_{ij}^t),$$

where the inequality follows from that $\lambda_0 \le O(\sqrt{1/(2wc^T \boldsymbol{H} c)})$ and $M = \sqrt{2c^T \boldsymbol{H} c / w}$. Let $\Delta := T^{-4}$, we have that, with probability at least $1 - T^{-4}$ and a fixed context i,

$$|\hat{c}_{ij}^t - c_{ij}| \le O(\sqrt{\log T} s_{ij}^t),$$

where $s_{ij}^t = \sqrt{g_{ij}^T (w\lambda_0 I_d + \sum_{\tau \in \mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T)^{-1} g_{ij}}$. Next, we show that with probability at least $1 - T^{-4}$, it holds that $s_{ij}^t \le O(\sqrt{d/n_j^t})$ with samples $n_j^t \ge \Omega(d \log(dT))$. Let $\hat{\Sigma} := w\lambda_0 I_d + \sum_{\tau \in \mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T$, and $\Sigma := w\lambda_0 I_d + n_j^t \mathbb{E}[gg^T]$, then we have,

$$\begin{aligned}(s_{ij}^t)^2 &= g_{ij}^T \hat{\Sigma}^{-1} g_{ij} \\ &\le |g_{ij}^T \hat{\Sigma}^{-1} (\Sigma - \hat{\Sigma}) \Sigma^{-1} g_{ij}| + g_{ij}^T \Sigma^{-1} g_{ij} \\ &\le O\left(\frac{d}{n_j^t}\right) \left\| \frac{1}{n_j^t} \sum_{\tau \in \mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T - \mathbb{E}[gg^T] \right\|_2 + O\left(\frac{d}{n_j^t}\right),\end{aligned} \tag{37}$$

where the first inequality follows from the triangle inequality, and the second from the bound on gradient $\|g\|_2 \le \sqrt{wL} \le \sqrt{d}$. Next, we need to bound the term $\| \frac{1}{n_j^t} \sum_{\tau \in \mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T - \mathbb{E}[gg^T] \|_2 \le O(1)$. Because, the gradients are bounded with $\|g\|_2 \le \sqrt{wL} \le \sqrt{d}$, with high probability $1 - T^{-4}$, we have the following inequality based on Theorem 1.6.2 (the Matrix Bernstein inequality) in (Tropp 2015),

$$\| \frac{1}{n_j^t} \sum_{\tau \in \mathcal{T}_j^t} g_{i_\tau a_\tau} g_{i_\tau a_\tau}^T - \mathbb{E}[gg^T] \|_2 \le \sqrt{\frac{2d \log(dT)}{n_j^t}} + \frac{\sqrt{d} \log(dT)}{3 n_j^t} \le O(1),$$

where the second inequality follows from the condition $n_j^t \ge \Omega(d \log(dT))$. Therefore, after taking the union bound for all context $i \in \mathcal{I}$ together with that c_{ij} can be perfectly predicted via the linear function, we obtain that, with probability at least $1 - \delta - T^{-4}$, it holds

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d \log T}{n_j^t}}\right),$$

where the inequality follows from the assumption that m is smaller than the polynomials of T. This concludes the proof of Proposition 2. □

Notice that for the neural network predictor, we require the effective dimension d (i.e., the number of parameters in the neural network) is polynomial in the number of contexts m, which is likely not tight. In fact, the dependence of error on d can be significantly reduced to a logarithmic dependence on the number of contexts m (Zhou et al. 2020, Tropp 2015).

G.2. Training with the Gradient Descent Algorithm

We now consider the gradient-based training procedure for neural networks to validate Assumption 2. In fact, one can devise a gradient descent training algorithm that achieves the same convergence rate as lazy training (i.e., $|\hat{c}_{ij}^t - c_{ij}| \le O(\sqrt{d \log T / n_j^t})$), because the training trajectory path, $\{\theta^t\}_{t=1}^T$, of the gradient descent procedure is close to that of lazy training. Formally, we propose the gradient-based training of a neural network as follows. This gradient descent procedure is an approximation of SGD. This training method can also be replaced by stochastic gradient descent with a more involved analysis such as Allen-Zhu et al. (2019).

Training a Neural Network with Gradient Descent at the Round t for ad j

Input: Step size η, number of gradient descent steps U, network width w, regularization parameter λ_0.

Loss function: $\mathcal{L}(\theta) := \sum_{\tau \in \mathcal{T}_j^t} (H_j(x_{i_\tau a_\tau}, \theta) - v_{i_\tau a_\tau})^2 / 2 + w\lambda_0 \|\theta - \theta_0\|_2^2 / 2$

For $u = 0, 1, 2, \ldots, U-1$ **do**

$\theta^{u+1} = \theta^u - \eta \nabla \mathcal{L}(\theta^u)$

The following proposition shows that Assumption 2 holds for a neural network if trained with the gradient descent algorithm described above.

PROPOSITION 3 **(Prediction Oracle with Gradient-based Training)**. *Under Assumption 3 and all the conditions of Proposition 2, for any ad $j \in A$, the predicted CTR at round t, $\hat{c}_{ij}^t$, is obtained by the gradient descent algorithm. For any δ, there exist a family of constants $\{C_i\}_{i=0}^5 > 0$ such that, if for all $t \in [T]$, the regularization parameter λ_0, training step size η, number of steps U, and network width w satisfy*

$$
\begin{aligned}
& w \geq C_0 m^4 L^6 \log(m^2 L/\delta)/\gamma^4 \\
& 2\sqrt{t/(w\lambda_0)} \geq C_1 w^{-3/2} L^{-3/2} [\log(mL^2/\delta)]^{3/2} \\
& 2\sqrt{t/(w\lambda_0)} \leq C_2 \min\{L^{-6}[\log w]^{-3/2}, (w(\lambda_0\eta)^2 L^{-6} t^{-1} (\log w)^{-1})^{3/8}\} \\
& \eta \leq C_3 (w\lambda_0 + twL)^{-1} \\
& U > C_4 \log(d\log(T))/\log(1-\eta w \lambda_0) \\
& w^{1/6} \geq C_5 \sqrt{\log w} L^{7/2} t^{7/6} \lambda_0^{-7/6} (1 + \sqrt{t/\lambda_0}),
\end{aligned}
$$

it holds that, if ad j is displayed $n_j^t \geq \Omega(d\log(dT))$ times and the random click-through outcomes $\{v_s(a_s) \in \{0,1\} : 1 \leq s \leq t\}$ are observed, then for all context $i \in \mathcal{I}$, with probability at least $1 - \delta - T^{-4}$, the following inequality holds:

$$
|\hat{c}_{ij}^t - c_{ij}| \leq O\left(\sqrt{\frac{d\log T}{n_j^t}}\right).
$$

Before proving Proposition 3, we first introduce Lemma 6 and Lemma 7 below to bound the training trajectory $\{\theta^t : t = 1, 2, ..., T\}$ and the gradient $\nabla_\theta H_j(x_{ij}, \hat{\theta})$, respectively.

LEMMA 6 **(Lemma B.2 in Zhou et al. (2020))**. *For any ad $j \in A$, there exist a family of constants $\{C_i\}_{i=1}^5 > 0$ such that for any $\delta \in (0,1)$, if for each $t \in [T]$, η and w satisfy*

$$
\begin{aligned}
& 2\sqrt{t/(w\lambda_0)} \geq C_1 w^{-3/2} L^{-3/2} [\log(mL^2/\delta)]^{3/2} \\
& 2\sqrt{t/(w\lambda_0)} \leq C_2 \min\{L^{-6}[\log w]^{-3/2}, (w(\lambda_0\eta)^2 L^{-6} t^{-1} (\log w)^{-1})^{3/8}\} \\
& \eta \leq C_3 (w\lambda_0 + twL)^{-1} \\
& w^{1/6} \geq C_4 \sqrt{\log w} L^{7/2} t^{7/6} \lambda_0^{-7/6} (1 + \sqrt{t/\lambda_0})
\end{aligned}
$$

then, with probability at least $1-\delta$ over the He Initialization of θ_0, we have, for any $t \in [T]$, $\|\theta^t - \theta_0\|_2 \leq 2\sqrt{t/w\lambda_0}$ and

$$
\|\theta^t - (A_j^t)^{-1} b_j^t - \theta_0\|_2 \leq (1 - \eta w \lambda_0)^{U/2} \sqrt{t/(w\lambda_0)} + C_5 w^{-2/3} \sqrt{\log w} L^{7/2} t^{5/3} \lambda_0^{-5/3} (1 + \sqrt{t/\lambda_0}). \tag{38}
$$

LEMMA 7 **(Lemma B.4 in Zhou et al. (2020))**. *For any ad $j \in A$, there exist a family of constants $\{C_i\}_{i=1}^3 > 0$ such that for any $\delta \in (0,1)$, if τ satisfies that*

$$
C_1 w^{-3/2} L^{-3/2} [\log(mL^2/\delta)]^{3/2} \leq \tau \leq C_2 L^{-6} [\log w]^{-3/2}.
$$

then, with probability at least $1-\delta$ *over the He Initialization of* θ_0*, for all* $\hat{\theta}$ *and* $\tilde{\theta}$ *satisfying* $\|\hat{\theta}-\theta_0\|_2 \le \tau$ *and* $\|\tilde{\theta}-\theta_0\|_2 \le \tau$*, we have, for any context* i*,*

$$\left|H_j(x_{ij},\tilde{\theta}) - H_j(x_{ij},\hat{\theta}) - \langle \nabla_\theta H_j(x_{ij},\hat{\theta}), \tilde{\theta}-\hat{\theta}\rangle\right| \le C_3\tau^{4/3}L^3\sqrt{w\log w}.$$

With Lemma 6 and Lemma 7, we are now ready to prove Proposition 3.

Proof of Proposition 3.

It suffices to consider the union of the high probability cases in Proposition 2, Lemma 6, and Lemma 7. Let us set $\tau = 2\sqrt{t/w\lambda_0}$ in Lemma 7. At round t, after observing n_j^t i.i.d samples of each ad j, for a fixed context i, we have the following inequality:

$$\begin{aligned}
|\hat{c}_{ij}^t - c_{ij}| &= |H_j(x_{ij},\theta^t) - \langle \nabla_\theta H_j(x_{ij},\theta_0), \theta^* - \theta_0\rangle| \\
&\le |H_j(x_{ij},\theta^t) - \langle \nabla_\theta H_j(x_{ij},\theta_0), (A_j^t)^{-1}b_j^t\rangle| + |\langle \nabla_\theta H_j(x_{ij},\theta_0), (A_j^t)^{-1}b_j^t\rangle - \langle \nabla_\theta H_j(x_{ij},\theta_0), \theta^* - \theta_0\rangle| \\
&\le |H_j(x_{ij},\theta^t) - \langle \nabla_\theta H_j(x_{ij},\theta_0), (A_j^t)^{-1}b_j^t\rangle| + O(\sqrt{d\log(T)/n_j^t}) \\
&\le |H_j(x_{ij},\theta^t) - H_j(x_{ij},(A_j^t)^{-1}b_j^t+\theta_0) + H_j(x_{ij},\theta_0)| + C_3\tau^{4/3}L^3\sqrt{w\log w} + O(\sqrt{d\log(T)/n_j^t}) \\
&\le |\langle \nabla_\theta H_j(x_{ij},\theta_0), -(A_j^t)^{-1}b_j^t - \theta_0 + \theta^t\rangle| + 2C_3\tau^{4/3}L^3\sqrt{w\log w} + O(\sqrt{d\log(T)/n_j^t}) \\
&\le (1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0} + O(t^{2/3}w^{-1/6}\lambda_0^{-2/3}L^3\sqrt{\log w}) + O(\sqrt{d\log(T)/n_j^t}),
\end{aligned} \tag{39}$$

where the equality follows from Lemma 5. The first inequality of (39) follows from the triangular inequality. The second inequality of (39) follows from Proposition 2. The third and fourth inequalities of (39) follow from Lemma 7, the fact that $\|(A_j^t)^{-1}b_j^t\|_2 \le \tau$ (see Lemma C.4 in Zhou et al. 2020), the triangular inequality, and $H_j(x_{ij},\theta_0) = 0$ by the *He Initialization* of θ_0. The last inequality of (39) follows from Lemma 6 which bounds $\|\theta^t - (A_j^t)^{-1}b_j^t - \theta_0\|_2$ using inequality (38), and the bound on the gradient $\|\nabla_\theta H_j(x_{ij},\theta_0)\|_2 \le \sqrt{wL}$ (see Cao and Gu 2019, Zhou et al. 2020). With a sufficiently large neural network width w, the second term of the last inequality (39), $O(t^{2/3}w^{-1/6}\lambda_0^{-2/3}L^3\sqrt{\log w})$, can be bounded by $O(\sqrt{d\log(T)/n_j^t})$. The first term of (39), $(1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0}$ converges to 0 at an exponential rate with respect to the number of training steps U. Because $U > \Omega(\log(d\log(T))/\log(1-\eta w\lambda_0))$, $(1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0}$ is also bounded by $O(\sqrt{d\log(T)/n_j^t})$. Following the same argument as the proof of Proposition 2, we obtain that, with probability $1-\delta-T^{-4}$, for any context i,

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d\log(1/\delta)}{n_j^t}}\right).$$

This concludes the proof of Proposition 3. □

Appendix H: Details of the Online Advertising System for Platform O

In this section, we describe the institutional details of the online advertising system for Platform. In particular, the auction mechanisms, billing options, and the PID system are introduced.

H.1. Auction Mechanisms and Billing Options

For a large scale online platform, the DSP allocates billions of ad impressions to hundreds of thousands of ads each day. In order not to ruin the user experience, the DSP needs to efficiently match the tremendous

number of ads and ad impressions within milliseconds. Before entering the auction stage, the DSP quickly downscales the size of the ad pool from hundreds of thousands to hundreds by simple filtering rules set by advertisers and predictive models. At the auction stage, hundreds of ads compete to win an ad impression based on advertisers' bids. The ad impression is allocated to the ad with the highest *estimated Cost per Mille* (eCPM) of the match, which measures the expected revenue of displaying the ad to the respective platform user for a thousand times. Such an allocation rule ensures that each ad impression generates the highest *ex ante* revenue in expectation.

The eCPM of a match between an ad and an ad impression depends on what the advertiser bids on (impression, click, or conversion). More specifically, if the advertiser bids on *impression*, eCPM is the bid itself (eCPM=bid). If the advertiser bids on *click*, eCPM equals the bid multiplied by the predicted CTR (pCTR) of the ad (eCPM=bid×pCTR). Finally, if the advertiser bids on *conversion*, eCPM equals the bid multiplied by the product of pCTR and the predicted conversion rate (pCVR) of the ad (eCPM=bid×pCTR×pCVR), where pCVR is defined as the rate of the user being converted after clicking the ad. Here, conversion means that, upon clicking an ad, a user eventually becomes the advertiser's customer. A typical conversion, sometimes also called an "action" of the user, may take different forms, such as app installation or deposit in a game.

Typically, there are several different billing options for advertisers to choose from, including *Cost per Mille* (CPM), *Cost per Click* (CPC), *Cost per Action* (CPA), *Optimized Cost per Mille* (oCPM) and *Optimized Cost per Click* (oCPC). We summarize the differences between these billing options in Table 11. Under the CPM, CPC, and CPA billing option, advertisers bid on the impressions, clicks, and conversions, respectively, and are directly charged after their ads are displayed, clicked, or converted. Due to the intrinsic uncertainty of user clicks and conversions, the advertiser bears a high risk under the CPM scheme. On the other hand, if an ad is displayed to a platform user, it already makes a negative impact on user experience and causes losses to the platform. As a consequence, in the current online advertising market (such as Facebook and Platform O), oCPM and oCPC under which the DSP and the advertiser *share* the click and conversion uncertainty risks are the most popular billing options. More specifically, under both oCPM and oCPC, the advertisers bid on conversion. However, the advertiser will be charged by the expected cost per impression, bid_conversion×pCTR×pCVR (resp. the expected cost per click, bid_conversion×pCVR), when the ad is displayed to (resp. clicked by) a user under oCPM (resp. oCPC). One should note that, because of the randomness in click-through and conversion rates, the actual payment of the advertiser per conversion is not necessarily the same as its bid for a conversion under oCPM or oCPC. For each billing option, the auction may run in a first-price or second-price fashion, under which the winning advertiser pays its own bid, or the bid with the second-highest eCPM.

H.2. PID-Based Bidding System

As introduced in Section 3, the PID controller is a feedback control device widely used in online advertising platforms, especially for the oCPC and oCPM billing options. The PID controller aims to gear the realized CPA of each ad as close to the target CPA as possible, so it increases the bid price thus boosting its eCPM

Table 11 Billing Options

Payment Scheme	Bid Price	Charged upon	Fee Deduction	eCPM (Rank by)
CPM	bid_impression	impression	bid_impression	bid_impression
CPC	bid_click	click	bid_click	bid_click×pCTR
oCPM	bid_conversion	impression	bid_conversion×pCTR×pCVR	bid_conversion×pCTR×pCVR
oCPC	bid_conversion	click	bid_conversion×pCVR	bid_conversion×pCTR×pCVR
CPA	bid_conversion	conversion	bid_conversion	bid_conversion×pCTR×pCVR

Note: This table is mainly for the first-price auction. Also, bid_impression, bid_click, bid_conversion are the bids on impressions, clicks, and conversions given by advertisers, respectively. The column Fee Deduction gives the budget depleted upon each impression, click, or conversion.

and the chance of winning the auction, if the actual cost per conversion of an ad falls below the target cost, and vice versa.

Both billing option oCPM and oCPC suffer from the issue that the actual cost (per conversion) of an advertiser is different from its bid (also known as the target cost). Such a cost-control issue is exacerbated by the following: (a) second-price auction, under which the winning advertiser pays the expected cost per impression/click of the bidder with the second-highest eCPM; and (b) biased estimation of pCTR and pCVR, under which for each ad and ad impression pair the DSP could not accurately estimate the CTR and CVR. In practice, the PID controller is widely adopted on online advertising platforms (Yang et al. 2019, Zhang et al. 2016) to control the gap between the actual cost and the target cost for an advertiser. Under PID, advertisers authorize the DSP to adaptively change their real-time bid prices to address the aforementioned *cost control problem*. The core of the PID controller is a simple feedback control idea: If the actual cost per conversion falls below the target cost, the DSP will increase the bid price thus boosting its eCPM and the chance of winning the auction, and vice versa. This real-time bid price given by the PID controller is referred to as the *System Bidding Price* throughout this paper.

Formally, the PID controller is formulated as Eq. (40).

$$
\begin{aligned}
&\mathsf{error}_t = \mathsf{targetBid} - \mathsf{realCost}_t/\mathsf{realConversion}_t,\\
&\mathsf{P}_t = k_p \times \mathsf{error}_t\\
&\mathsf{I}_t = k_i \times \sum_{t' \le t} \mathsf{error}_{t'},\\
&\mathsf{D}_t = k_d \times (\mathsf{error}_t - \mathsf{error}_{t-1})\\
&\mathsf{PID}_t = \mathsf{P}_t + \mathsf{I}_t + \mathsf{D}_t,\\
&\mathsf{systemBid}_{t+1} = \mathsf{systemBid}_t + \mathsf{PID}_t \times \mathsf{systemBid}_t
\end{aligned}
\tag{40}
$$

It is clear from the above formulation that the PID controller changes the system bidding price $\mathsf{systemBid}_t$ after accumulating the feedback data for each ad within a fixed amount of time. The first equation quantifies the gap between the target cost and the real cost (the total actual cost divided by the total actual conversions). P_t, I_t, D_t represents the Proportional, Integral, Derivative (PID) term in the PID system respectively. And the corresponding coefficients k_p, k_i, k_d are hyper-parameters to be fine-tuned. Readers interested in more details about the PID system are referred to, e.g., Yang et al. (2019) and Zhang et al. (2016).